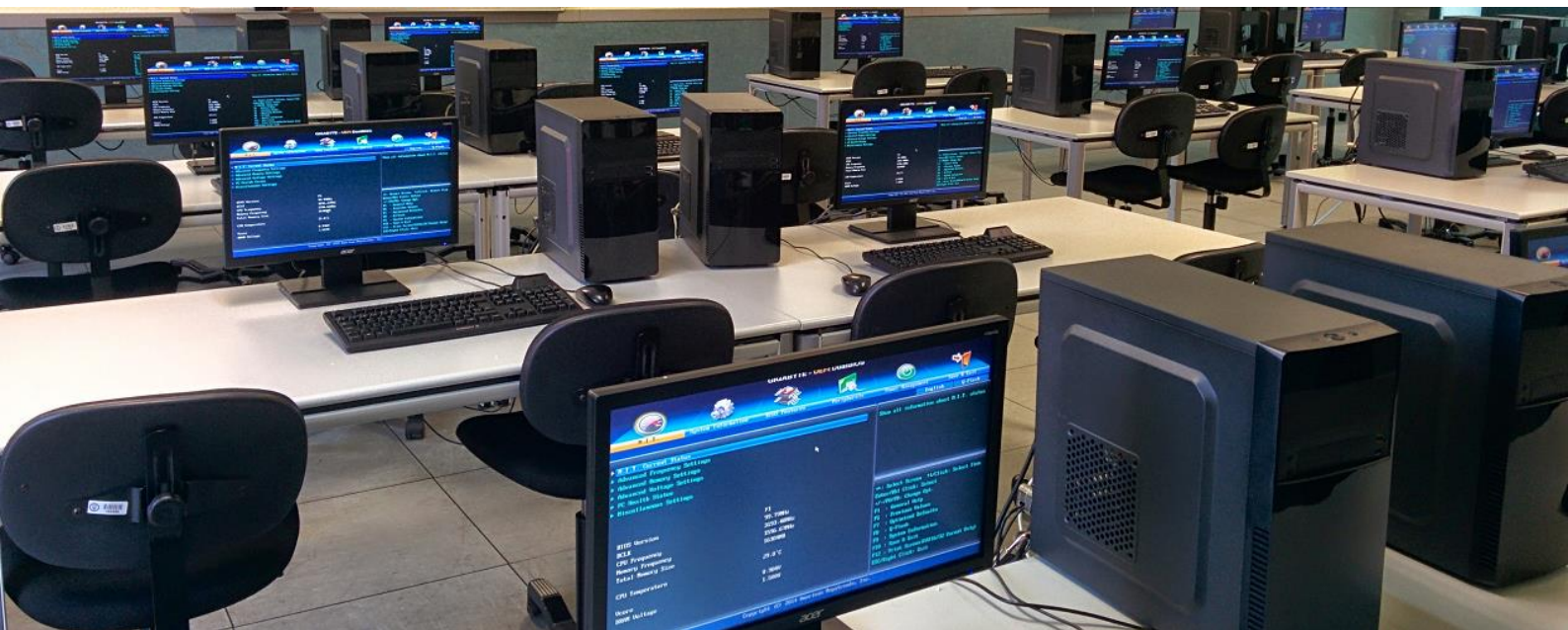




Autor: Francisco Javier Rodríguez Isabel | Tutor: Alejandro Calderón Mateos

Grado en Ingeniería Informática



Trabajo de Fin de Grado

Análisis, diseño e implantación de una aplicación para la gestión de reservas para las aulas docentes del departamento de informática

Índice de contenidos

Índice de Figuras.....	5
Índice de Tablas	7
Agradecimientos.....	12
Parte en Inglés	13
Introduction	13
Motivation.....	13
Objectives.....	13
Rest of this document.....	14
Abstract.....	15
Conclusions and Future Work	22
Conclusions.....	22
Future Work.....	24
1. Introducción.....	26
1.1. Motivación.....	26
1.2. Objetivos	26
1.3. Resto del documento	27
2. Estado de la cuestión	28
2.1. Estado del arte	28
2.2. <i>Content Management System</i>	29
Drupal	29
WordPress.....	31
Joomla	32
Aplicaciones similares al gestor de reservas	33
Módulos.....	33
Plugins	34
Extensiones	34
2.4. APIs y otras herramientas	35
2.5. Entorno del servidor	35

3. Análisis	38
3.1. Introducción.....	38
3.2. Valoración de las alternativas.....	39
<i>Content Management System</i>	39
<i>Plugins</i>	41
Selección del entorno operacional	42
3.3. Definición de los requisitos del sistema	43
Requisitos de Capacidad	45
Requisitos de Restricción.....	51
4. Diseño	54
4.1. Definición de la arquitectura del sistema.....	54
Especificación de las excepciones	56
4.2. Diseño de la arquitectura de soporte.....	57
4.3. Diseño de casos de uso	61
4.4. Funcionamiento y taxonomía de la interfaz.....	63
Pantalla de ocupación diaria	64
Pantalla de ocupación por aulas.....	66
Pantalla de login	68
Pantalla de solicitud de reservas.....	69
Pantalla de gestión de reservas.....	71
Pantalla de creación de reservas.....	72
Pantalla de gestión de aulas.....	74
Pantalla de gestión de asignaturas	75
Pantalla de gestión de grados	76
Pantalla de gestión de software.....	77
Pantalla de gestión de máquinas virtuales.....	78
Pantalla de gestión de informes	79
Pantalla de opciones	80
4.5. Diseño de clases	82

4.6. Diseño físico de datos.....	100
5. Implementación e Implantación.....	104
5.1. Establecimiento del plan de implantación.....	104
5.2. Formación necesaria.....	105
5.3. Elaboración de los manuales.....	105
5.4. Incorporación del sistema al entorno de operación.....	106
5.5. Carga de datos al entorno de operación.....	106
5.6. Preparación del mantenimiento del sistema.....	107
Entorno de pruebas.....	107
Entorno de producción.....	107
5.7. Presentación y aprobación del sistema.....	108
5.8. Paso a producción.....	108
6. Evaluación.....	110
6.1. Plan de Pruebas.....	110
Pruebas unitarias.....	110
Pruebas de integración.....	121
Pruebas de implantación del sistema.....	127
Pruebas de aceptación del sistema.....	128
6.2. Satisfacción de usuarios.....	128
6.3. Matrices de trazabilidad.....	133
Matriz de trazabilidad entre requisitos y pruebas unitarias.....	133
Matriz de trazabilidad entre requisitos y pruebas de integración.....	136
Matriz de trazabilidad entre requisitos y pruebas de implantación.....	138
7. Planificación y presupuesto.....	139
7.1. Planificación.....	139
7.2. Presupuesto.....	140
Salarios.....	141
Amortizaciones.....	142
Costes.....	143

Precio final del proyecto	144
7.3. Esfuerzo Estimado vs Esfuerzo Real	144
8. Conclusiones y trabajos futuros	148
8.1. Conclusiones.....	148
Producto	148
Proceso.....	149
Personales	149
8.2. Trabajos futuros.....	150
Apéndice I: Manual de administración	152
Instalar la aplicación.....	152
Configuración.....	152
Gestión de reservas	154
Shortcodes.....	155
Bibliografía	158
Acrónimos, abreviaturas y definiciones.....	160

Índice de Figuras

Figura 1: Método de trabajo (Inglés).....	17
Figura 2: Planificación (Inglés)	18
Figura 3: Equipo de trabajo (Inglés).....	19
Figura 9: Página de inicio de Drupal.....	30
Figura 10: Panel de administración de Drupal.....	30
Figura 11: Página de inicio de WordPress	31
Figura 12: Página de administración de WordPress	31
Figura 13: Página de inicio de Joomla	32
Figura 14: Página de administración de Joomla	33
Figura 15: Modelo Vista Controlador	54
Figura 16: Caso de Uso CU_01	61
Figura 17: Caso de Uso CU_02.....	61
Figura 18: Caso de Uso CU_03.....	62
Figura 19: Caso de Uso CU_04.....	62
Figura 20: Caso de Uso CU_05.....	63
Figura 21: Pantalla de ocupación diaria	64
Figura 22: Pantalla de ocupación por aulas.....	66
Figura 23: Pantalla de login	68
Figura 24: Pantalla de solicitud de reservas 1	69
Figura 25: Pantalla de solicitud de reservas 2	69
Figura 26: Pantalla de solicitud de reservas 3	70
Figura 27: Pantalla de gestión de reservas.....	71
Figura 28: Pantalla de creación de reservas 1	72
Figura 29: Pantalla de creación de reservas 2	72
Figura 30: Pantalla de creación de reservas 3	73
Figura 31: Pantalla de gestión de aulas.....	74
Figura 32: Pantalla de gestión de asignaturas	75
Figura 33: Pantalla de gestión de grados	76
Figura 34: Pantalla de gestión de software.....	77
Figura 35: Pantalla de gestión de máquinas virtuales.....	78
Figura 36: Pantalla de gestión de informes.....	79
Figura 37: Pantalla de opciones	80
Figura 38: Diagrama de clases	84
Figura 39: Esquema relacional	101
Figura 40: Planificación.....	139

Figura 41: Planificación estimada	144
Figura 42: Planificación real	145
Figura 43: Coste de personal.....	146

Índice de Tablas

Tabla 1: Coste del proyecto (Inglés)	20
Tabla 3: Alternativa de servidores 1	36
Tabla 4: Alternativa de servidores 2	36
Tabla 5: Alternativa de servidores 3	37
Tabla 6: Comparativa de CMS	41
Tabla 7: Comparativa de Plugins	42
Tabla 8: Requisitos de ejecución	43
Tabla 9: Ejemplo de tabla de requisitos	43
Tabla 10: Requisito UR-C001	45
Tabla 11: Requisito UR-C002	45
Tabla 12: Requisito UR-C003	45
Tabla 13: Requisito UR-C004	45
Tabla 14: Requisito UR-C005	46
Tabla 15: Requisito UR-C006	46
Tabla 16: Requisito UR-C007	46
Tabla 17: Requisito UR-C008	46
Tabla 18: Requisito UR-C009	47
Tabla 19: Requisito UR-C010	47
Tabla 20: Requisito UR-C011	47
Tabla 21: Requisito UR-C012	47
Tabla 22: Requisito UR-C013	48
Tabla 23: Requisito UR-C014	48
Tabla 24: Requisito UR-C015	48
Tabla 25: Requisito UI-C016	48
Tabla 26: Requisito UI-C017	49
Tabla 27: Requisito UI-C018	49
Tabla 28: Requisito UI-C019	49
Tabla 29: Requisito UI-C020	49
Tabla 30: Requisito UI-C021	50
Tabla 31: Requisito UI-C022	50
Tabla 32: Requisito UI-C023	50
Tabla 33: Requisito UI-C024	50
Tabla 34: Requisito UI-C025	51
Tabla 35: Requisito UK-R001	51
Tabla 36: Requisito UK-R002	51

Tabla 37: Requisito UR-R003.....	51
Tabla 38: Requisito US-R004.....	52
Tabla 39: Requisito UR-R005.....	52
Tabla 40: Requisito UI-R006.....	52
Tabla 41: Requisito UI-R007.....	52
Tabla 42: Requisito UI-R008.....	53
Tabla 43: Excepción EX-001.....	56
Tabla 44: Excepción EX-002.....	56
Tabla 45: Excepción EX-003.....	56
Tabla 46: Excepción EX-004.....	57
Tabla 47: Subsistema SSV-01 (Módulo de calendario).....	57
Tabla 48: Subsistema SSM-02 (Módulo de acceso a la información).....	58
Tabla 49: Subsistema SSC-03 (Módulo de usuario).....	58
Tabla 50: Subsistema SSC-04 (Módulo de administración).....	58
Tabla 51: Subsistema SSV-05 (Módulo de gestión de reservas).....	59
Tabla 52: Subsistema SSV-06 (Módulo de creación de reservas).....	59
Tabla 53: Subsistema SSV-07 (Módulo de gestión de aulas).....	59
Tabla 54: Subsistema SSV-08 (Módulo de gestión de asignaturas).....	59
Tabla 55: Subsistema SSV-09 (Módulo de gestión de grados).....	60
Tabla 56: Subsistema SSV-10 (Módulo de gestión de software).....	60
Tabla 57: Subsistema SSV-11 (Módulo de gestión de máquinas virtuales).....	60
Tabla 58: Subsistema SSV-12 (Módulo de gestión de informes).....	60
Tabla 59: Subsistema SSV-13 (Módulo de gestión de opciones).....	61
Tabla 60: Caso de Uso CU_01.....	61
Tabla 61: Caso de Uso CU_02.....	61
Tabla 62: Caso de Uso CU_03.....	62
Tabla 63: Caso de Uso CU_04.....	62
Tabla 64: Caso de Uso CU_06.....	63
Tabla 65: Pantalla de ocupación diaria.....	65
Tabla 66: Pantalla de ocupación por aulas.....	67
Tabla 67: Pantalla de login.....	68
Tabla 68: Pantalla de solicitud de reservas.....	70
Tabla 69: Pantalla de gestión de reservas.....	71
Tabla 70: Pantalla de creación de reservas.....	73
Tabla 71: Pantalla de gestión de aulas.....	74
Tabla 72: Pantalla de gestión de asignaturas.....	75
Tabla 73: Pantalla de gestión de grados.....	76

Tabla 74: Pantalla de gestión de software.....	77
Tabla 75: Pantalla de gestión de máquinas virtuales	78
Tabla 76: Pantalla de gestión de informes	79
Tabla 77: Pantalla de gestión de informes	81
Tabla 78: Clases de lógica de calendario	82
Tabla 79: Clases de acceso a la información.....	82
Tabla 80: Clases de lógica de usuario	82
Tabla 81: Clases de lógica de administración.....	82
Tabla 82: Clases de lógica de gestión de reservas	82
Tabla 83: Clases de lógica de creación de reservas	82
Tabla 84: Clases de lógica de gestión de aulas	82
Tabla 85: Clases de lógica de gestión de asignaturas.....	82
Tabla 86: Clases de lógica de gestión de grados.....	83
Tabla 87: Clases de lógica de gestión de software	83
Tabla 88: Clases de lógica de gestión de máquinas virtuales	83
Tabla 89: Clases de lógica de gestión de informes	83
Tabla 90: Clases de lógica de gestión de opciones.....	83
Tabla 91: Clase Calendar	86
Tabla 92: Clase Database	89
Tabla 93: Clase User.....	89
Tabla 94: Clase Administrator	91
Tabla 95: Clase Gest_Schedule.....	92
Tabla 96: Clase Create_Schedule.....	94
Tabla 97: Clase Gest_Classroom.....	95
Tabla 98: Clase Gest_Course.....	96
Tabla 99: Clase Gest_Degree	97
Tabla 100: Clase Gest_Software.....	98
Tabla 101: Clase Gest_VM.....	98
Tabla 102: Clase Gest_Data	99
Tabla 103: Clase Gest_Options.....	100
Tabla 104: Clase Confirm_Schedule	100
Tabla 105: Tabla reservas_opciones.....	101
Tabla 106: Tabla reservas_aulas.....	101
Tabla 107: Tabla reservas_maquinas_virtuales.....	102
Tabla 108: Tabla reservas_software.....	102
Tabla 109: Tabla reservas_grado	102
Tabla 110: Tabla reservas_asignatura	103

Tabla 111: Tabla reservas_mv_asignatura	103
Tabla 112: Tabla reservas_sw_asignatura.....	103
Tabla 113: Tabla reservas_reservas.....	103
Tabla 114: Prueba PU-N001	111
Tabla 115: Prueba PU-N002	111
Tabla 116: Prueba PU-N003	112
Tabla 117: Prueba PU-N004	112
Tabla 118: Prueba PU-N005	112
Tabla 119: Prueba PU-N006	113
Tabla 120: Prueba PU-N007	113
Tabla 121: Prueba PU-N008	113
Tabla 122: Prueba PU-N009	114
Tabla 123: Prueba PU-N010	114
Tabla 124: Prueba PU-N011	114
Tabla 125: Prueba PU-N012	115
Tabla 126: Prueba PU-N013	115
Tabla 127: Prueba PU-N014	115
Tabla 128: Prueba PU-N015	116
Tabla 129: Prueba PU-N016	116
Tabla 130: Prueba PU-N017	116
Tabla 131: Prueba PU-N018	117
Tabla 132: Prueba PU-N019	117
Tabla 133: Prueba PU-N020	117
Tabla 134: Prueba PU-N021	118
Tabla 135: Prueba PU-N022	118
Tabla 136: Prueba PU-N023	118
Tabla 137: Prueba PU-N024	119
Tabla 138: Prueba PU-N025	119
Tabla 139: Prueba PU-N026	119
Tabla 140: Prueba PU-N027	120
Tabla 141: Prueba PU-N028	120
Tabla 142: Prueba PU-N029	120
Tabla 143: Prueba PI-001.....	121
Tabla 144: Prueba PI-002.....	121
Tabla 145: Prueba PI-003.....	122
Tabla 146: Prueba PI-004.....	122
Tabla 147: Prueba PI-005.....	122

Tabla 148: Prueba PI-006.....	123
Tabla 149: Prueba PI-007.....	123
Tabla 150: Prueba PI-008.....	123
Tabla 151: Prueba PI-009.....	124
Tabla 152: Prueba PI-010.....	124
Tabla 153: Prueba PI-011.....	124
Tabla 154: Prueba PI-012.....	125
Tabla 155: Prueba PI-013.....	125
Tabla 156: Prueba PI-014.....	125
Tabla 157: Prueba PI-015.....	126
Tabla 158: Prueba PI-016.....	126
Tabla 159: Prueba PI-017.....	126
Tabla 160: Prueba PI-018.....	127
Tabla 161: Prueba PIS-001	127
Tabla 162: Prueba PIS-002	127
Tabla 163: Prueba PIS-003	128
Tabla 164: Matriz de trazabilidad 1.....	133
Tabla 165: Matriz de trazabilidad 2.....	134
Tabla 166: Matriz de trazabilidad 3.....	135
Tabla 167: Matriz de trazabilidad 4.....	136
Tabla 168: Matriz de trazabilidad 5.....	137
Tabla 169: Matriz de trazabilidad 6.....	138
Tabla 170: Cálculo de costes empresariales.....	141
Tabla 171: Costes de equipos	142
Tabla 172: Amortizaciones	142
Tabla 173: Material fungible.....	142
Tabla 174: Gastos y Presupuesto	143
Tabla 175: Precio del Proyecto	144
Tabla 176: Precio real	147
Tabla 177: Precio real vs. estimado	147

Agradecimientos

Me gustaría mostrar mi agradecimiento a mi familia, por haberme apoyado en todos mis años de universidad, especialmente a mis padres por su confianza en mis capacidades y a mi hermana por mostrarme el camino a seguir.

A mis compañeros de universidad, por haberme acompañado durante toda la carrera. En especial a Miguel Ángel, sin cuya competencia sana no habría llegado a donde estoy hoy.

A los técnicos de laboratorio Óscar, Jaime y Roberto que completaron mi formación académica gracias a las prácticas en el laboratorio de informática donde adquirí lo necesario para desenvolverse en el mundo laboral. Y a mis compañeros de beca que me enseñaron la importancia del trabajo en equipo y las relaciones profesionales.

A mi tutor Alejandro que confió en mis capacidades para realizar un proyecto de esta envergadura y a todos los profesores de la universidad que con su esfuerzo y dedicación me han enseñado lo necesario para convertirme en ingeniero informático.

A todos ellos, gracias.

Parte en Inglés

Introduction

Motivation

The project **Análisis, diseño e implantación de una aplicación para la gestión de reservas para las aulas docentes del departamento de informática** has been proposed by the Laboratory of the Computer Science and Engineering Department. Our Laboratory needs to update the current website to empower it.

The most important tool included in the current website is the reservation tool. This tool needs to be updated too, especially its reservation manager which has the ability to manage the usage of classrooms for both, teachers and students.

This existing reservation manager was developed by editing a Drupal source code. It is hardcoded. This is the reason why **the website could not be updated**: the Drupal updating entailed an overwrite process on the reservation manager.

In order to resolve these issues, I was proposed to **design a new website for the laboratory** by employing modern APIs and CMS. The result would be a new reservation manager which facilitates the updating process of the system. This new reservation manager will be easier to maintain and it will also include new features which will make the work of technicians more manageable.

Objectives

The goal of this project is to upgrade the laboratory website by developing a **new reserve management** that includes new features in order to make the work of technicians easier. Thanks to this work, we will be able to:

- **Manage all classrooms** as well as their reservation requests with a new responsive interface based on HTML5 and JavaScript.
- **Check the software and virtual machines** which are necessary for each course automatically.

- **Allow teachers to request classrooms** for their courses and to send mails to technicians in order to verify the reservation.
- **Generate statistics** about classroom usage and to export this information in a Excel compatible format.

Rest of this document

This memory details the design and development stages of this project. It is structured as follows:

1. **Introduction.** It includes the motivations and objectives expected to be achieved during the development process.
2. **State of the Art.** It explains the current situation regarding the management of reserves and some information about the commercial solutions that could be useful for my project.
3. **Analysis.** This section explores some alternatives defined in the previous part (state of art). They will be compared in order to determine the best one to be used for this project. It will be also established the project scope with the client, and I also includes the first stages of its design.
4. **Design.** Here it will be discussed the technical design of the module to be designed and developed, and the choices made during the development.
5. **Implementation and Implantation.** This part deals with the installation process and the testing plan which guarantees a correct performance and the best possible quality of the project.
6. **Evaluation.** Here it will be defined a test session that the system should pass before releasing the 1.0 version.
7. **Planning and Budget.** It includes the planning for this project as well as its developing cost (in money).
8. **Conclusions and Future Work.** It will be explained the impressions after having developed the project and some future ideas to be included in the future.

Finally, this paper will provide a specialized bibliography on the subject. Furthermore, there is also a final appendix in which it is included an **administration manual** to teach technicians from the laboratory how to work with it.

Abstract

This project will demonstrate my personal skills with regard to the design and development of a **complete computer engineering project** as the final part of my academic training.

Gestor de Reservas is a web application which helps administrators from the computer laboratory to manage every classroom of the laboratory.

This project fills the need of updating the website and of a new reservation manager. The final result will be an improved system with some new features which will make easier its usage.

Gestor de Reservas will have the following characteristics:

- **Reservation view:** the system allows users to know the reservations for each day or for one specific classroom.
- **Reservation requests:** the system allows teachers to book classrooms in available periods.
- **Manage reserves:** the technicians from the computer laboratory will be able to manage every reserve from an administrator panel.

This application will be easier and more comfortable for students, teachers and technicians. A simpler and modern interface will make easier its employment in smartphones and tablets. It will be more comfortable since every step to manage reserves will be provided automatically by the app.

Firstly, it will be **analyzed the most popular CMS alternatives** to update the website. Secondly it will be sought some extensions that simplify the developing process. Among all CMS alternatives, it will be studied the following:

1. **Drupal.** Drupal is an open source tool which embodies a management platform powering millions of websites and applications. It is the current CMS used by technicians from the laboratory as well as the original reserve manager of the website.
2. **WordPress.** WordPress is a CMS software which is able to create an attractive website or blog. It is one of the most popular CMS and it will be studied the possibility of using it with some plugins like The Events

Calendar or Weekly Class Schedule to successfully develop the new Gestor de Reservas.

3. **Joomla.** Joomla is an award-winning CMS, which enables you to create websites and powerful online applications. Joomla is a complete suite and it could be used to develop that new website thanks to the Scheduler or Schedule School extensions.

To evaluate these options I will use the following criteria:

- a) **Design.** The design should be responsive, attractive, and functional as well as it should follow the modern design guidelines.
- b) **Performance.** The performance of the chosen CMS should guarantee a correct working with the server specifications.
- c) **Functionality.** The chosen CMS should be able to carry out every function that this system must produce.
- d) **Easy to use.** The new system should be easier than the current one.
- e) **Maintenance.** The maintenance of the system must be as easy as possible.
- f) **Content.** It is necessary that the chosen CMS allow adding new content from other developers.
- g) **Make content.** It must be possible to create new content when necessary.

After analyzing, it will be discovered that **no solution is possible** to implement all these functionalities on a CMS. Therefore, it will be necessary to develop a new web application.

It has been decided that this application will be **developed as a plugin of WordPress**. Furthermore, it will be taken the JTable API to simplify its development, as well as a PHP generator which can create an excel document that download some statistics about the booking done.

The **requirements for the application server** must be the same than the original one. The original hardware has a 8GB RAM and 500GB hard disk server, with Debian Jessie as operating system. Within this server there is a virtual machine in which the system is installed. The requirements for the virtual machine are: 2GB RAM (512MB swap), 20GB HDD and Debian Jessie.

Next sections of the project introduce the technical design that includes the definition of the system requirements with a table which specifies the "Capacity Requirements" (what the application should be able to do; for instance: to make a new reservation) and the "Restriction Requirements" (how the application should do that; for instance: to guarantee the security on data).

It will also be defined the subsystem integration in the application as well as the use cases of the three roles which are able to use the system: technicians (administrator privileges), teacher (capability to request reserves) and university students (only read permissions).

The **working method** followed for the development of this software is based on a six-step cycle in which each step must be reviewed before going on:

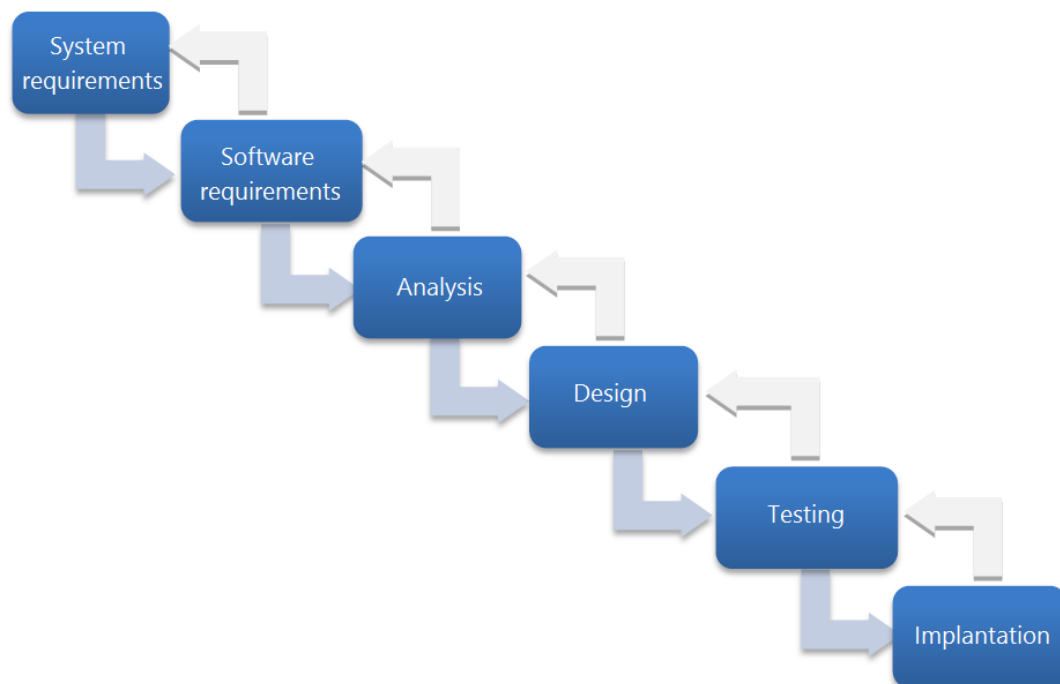


Figura 1: Método de trabajo (Inglés)

The first step deals with the analysis of the client objectives and the software requirements which the application must have implemented. Then, the system will be defined in deep before creating the app in the "Design" step. The last steps deal with a testing plan and the implantation process.

During the analysis part, it will be selected the better alternatives and define the first features which will be included in the project. **The architecture**

system is based on a MVC (Model View Controller) with up to 13 subsystems each one playing one function of the system.

Regarding the interface of the system designed, it will be created a responsive web which follows the guidelines of a WordPress design in the administration panel. The style will be similar to the one existing in the Carlos III University website.

It is also important to properly design how data is stored and backup. Due to the data to be used is other critical aspect of the project, we have to manage it carefully. And the proposed design has to deal with it. The solution is based on a database. This database contains fourteen tables which embody important information regarding reserves, classrooms, degrees, courses and so on.

The TFG syllabus establishes duration of between 300 and 360 hours for its development. If the team works 20 hours per week, the duration of this project will be around 18 weeks. In order to coordinate the project making with the exam period and holydays, this is the **final schedule**:

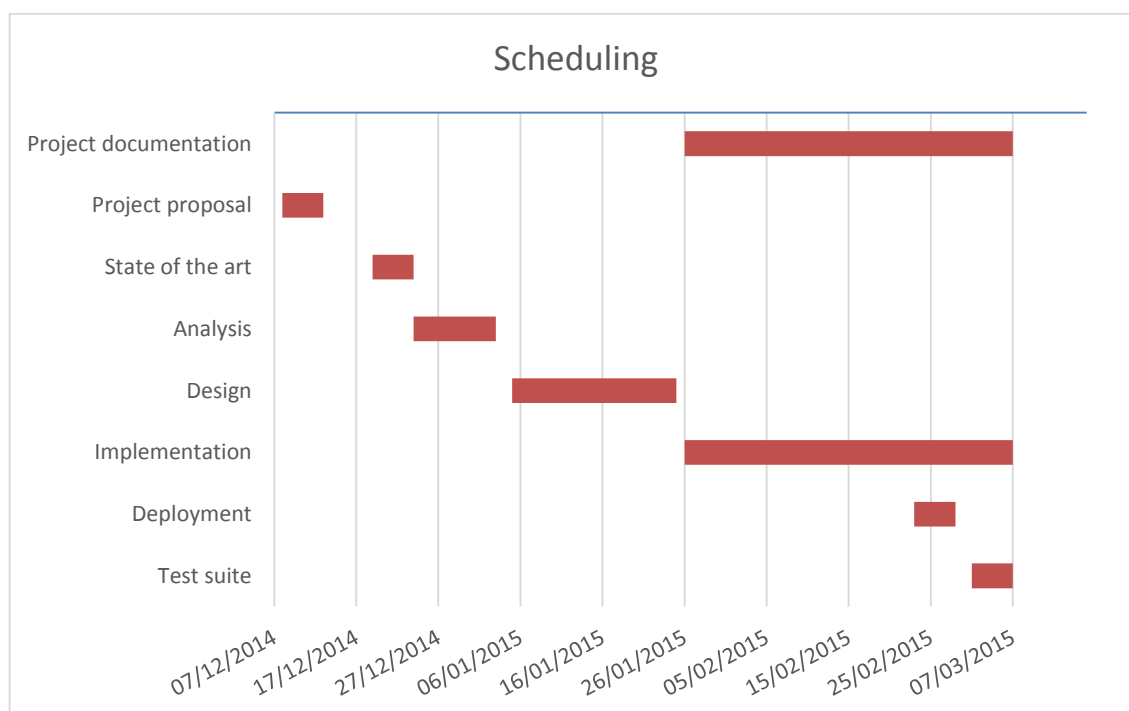


Figura 2: Planificación (Inglés)

The **working team** necessary to create the proposed system is composed by a system administrator, a system analyst and a web developer. All must be working as part-time developer for the system, and their work depends on what customer needs.

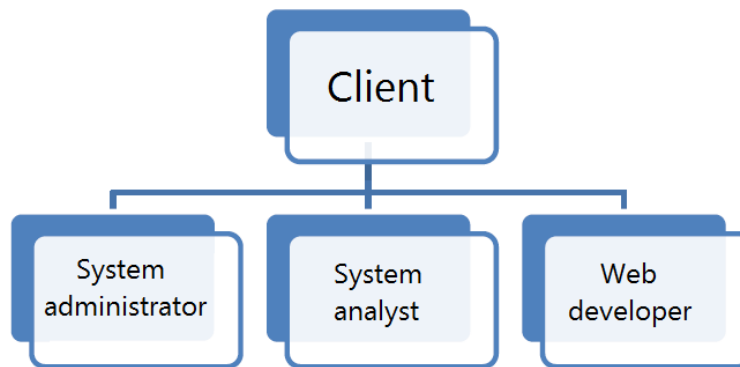


Figura 3: Equipo de trabajo (Inglés)

In order to guarantee the quality of the project, some **control sessions** have been scheduled in order to allow technicians from laboratory to supervise each stage of development. In addition, every document will be sent to technicians to have the design and manuals checked.

To ensure the quality of the application it has also created a complete plan which **tests the following parts**: unit, integration, implantation, acceptance and stakeholders' satisfaction test. This method will provide a useful feedback to improve the system and discover new features which could be included in the future. It is of vital importance that every part of the test results passed in order to release the system for production.

The **software used** in this project includes Microsoft Office Suite (for documentation), Google Keep (to organize the production) and OneDrive (to backup and sync the new versions with all devices connected with this development). NetBeans is the IDE chosen to develop the code, and PuTTY for performs remote operations.

In next section the cost of the application is studied. The estimated final cost of application is **22.114,39 € (TWENTY TWO THOUSAND ONE HUNDRED FOURTEEN EUROS THIRTY NINE CENTS)**. This price includes a backwardation that guarantees the support for three month after releasing the system version 1.0 for production.

Project Price
Business costs
16.186,78 €
Indirect costs (8%)
1.294,94 €
Blackwardation (10%)
1.748,17 €
Profit margin (15%)
2.884,49 €
Final Price
22.114,39 €

Tabla 1: Coste del proyecto (Inglés)

Finally, this document includes conclusions obtain during making this project and the possible future works. This document includes an administrator manual for technicians learn use, install and configure the application in any WordPress installation.

Definitely, this project is a proof that I acquired the knowledge necessary to prove that I am a computer science engineer. I used **knowledge from courses like** "Sistemas Operativos", "Ficheros y Bases de Datos" and "Diseño de Sistemas Operativos" for configure the server; from "Ingeniería del Software" and "Dirección de Proyectos de Desarrollo del Software" for documentation and knowledge from "Programación" and "Estructura de Datos y Algoritmos" for create algorithms and the code necessary to make a professional solution that the laboratory will be used for many years without the problems that they had with the previous solution.

But I also had learn news technologies, and had used resources that I had learn by myself like WordPress plugin develop, PHP coding, server manage and virtual machine configuration. This learning process should not possible without the previous formation that I received in the University Degree.

I'm excited to announce that my project will be published on the laboratory web page, and all university students and teachers will used to show the laboratory services offer while technicians simplify your work thanks to the application that I developed.

You can view the result on the project on web laboratory in (when release it):

<http://www.lab.inf.uc3m.es>

Also you can download the source code of the manage reserves app from this link:

<http://1drv.ms/1DQ8Tw4>

Conclusions and Future Work

Conclusions

Product: The Work Final Degree Project in this document (*Análisis, diseño e implantación de una aplicación para la gestión de reservas para las aulas docentes del departamento de informática*) has been the technical design for managing classroom from the Laboratory of Computer Science Department at the Carlos III University. This software has been developed as a WordPress plugin can work with any version up to 4.0.

The beginning of the document marks some objectives that the application must be complying for satisfice the client requirements. We can say the objectives were completed because:

- **The management process** is now easier than the original application and the new interface is mobile compatible.
- **It is possible to manage not only reserves on classroom**, but show information. The system is able to include information about degrees, courses, software and virtual machines to offer to technicians an easy management of all facets related with the classrooms.
- **Teachers can request reservations** with a form protected by the LDAP authentication system from the University.
- **Technicians can see statistics** about classroom booking and will be download it.

For ensure the objectives has been success, it is explained some samples about application working:

Also, at the beginning of the develop process, the version of WordPress was 4.0. Now, the newest version is 4.2. The update process not required any changes on the system. This demonstrated that application developed is compatible and let administrators an easier, **comfortable, and smoother update process**.

Process: The developing process couldn't be realized without the knowledge acquired during the degree.

The most significant issue is the delay of four week, explained in section 7.3. *Esfuerzo Estimado vs Esfuerzo Real*. This delay is due to a poor estimate time needed for learning some of the new technologies are need.

This delay don't cause real problems of deadlines and presentation, but the project price increase of 3.465,91 €, above the safety margin of 10%. This resulted in a reduction of the benefits of the project.

A good scheduling is very important. In this case **our scheduling** we will be able to obtain benefits, but a little error **cause a big increased cost**. Lack of experience is one of the reasons for error in the scheduling, but is a valuable lesson to learn for future projects.

Personal: I am very proud of this project. It not only demonstrated my capabilities as computer science engineer, but also shows that I can schedule a project and learn new technologies independently.

Especially important for this project are the **contents of courses like** "Interfaces de Usuario", "Tecnologías Informáticas para la Web" for web develop part; "Ingeniería de la Seguridad" and "Principios de Desarrollo del Software" for testing; "Sistemas Operativos", "Sistemas Distribuidos", "Diseño de Sistemas Operativos" and "Ficheros y Bases de Datos" for the server part; or "Ingeniería del Software" and "Dirección de Proyectos de desarrollo del Software" for generate documentation.

I need learning new thinks for complete this project. I need learned PHP language and acquire extra knowledge about Linux, virtualization and server administration.

I am particularly proud that my project is to be used in the University, because for me it is recognition of the system quality. Also, I am happy to facilitate the technicians of Laboratory of Department of Computer Science because they always helped me in my training. Help them is an extra satisfaction that joins have achieved all.

“Gestor de Reservas” is a great application and I’m sure that it will be used for many years into the laboratory of the computer department. With my contribution I can help to the laboratory that helped me during my university period.

Future Work

At last, I would like propose some ideas about future features that could be implement in the application:

1. **Accessibility.** Be sure it follows all accessibilities recommendations from main standards.
2. **More personalization for a commercial use.** The application is designed for the laboratory of computer engineering in the university, but could be possible apply some changes like tables customization for offer the project like a commercial solution that any organization can be used for your classroom management (gyms, schools and others university departments for example).
3. **Including incident and equipment managing.** The application can be part of a center classroom management tool if it includes the possibility for registering incidents and problems on the PCs, and develop a system to register the available equipment in each classroom like headphones, mouse and keyboards.
4. **More powerful statistics system.** The developed system include a statistics generator with all necessary information that laboratory must be send to department about the classroom occupation. However, will be a good idea implements new statistics for internal use with information about favorite classrooms, more frequently courses and others.
5. **Mobile application.** The developed system has a customizable web interface for smartphone and tablets, but use a website for work will be hard for technicians. It is possible developing application for Android or iOS focuses only on manage classrooms.
6. **Information panels for students.** Because all information about classroom is now centralized in the application, it might be interesting to include panels in the classroom that will show: occupation, available software, etc. This panel will be a monitor connected to any type of board like Raspberry Pi or Arduino.

7. **Manage computers.** The objective of application is centralizing all work about classroom. It is a good idea integrates other tools developed by laboratory like the scripts for start/reboot the classroom computer remotely. These scripts may be a web interface within this application.

1. Introducción

1.1. Motivación

Análisis, diseño e implantación de una aplicación para la gestión de reservas para las aulas del departamento de informática es un proyecto propuesto por el Laboratorio del Departamento de Informática basado en la necesidad de actualizar el actual sitio web del laboratorio.

Esta web no solo contiene información acerca del laboratorio, también incluye **varias herramientas indispensables para el trabajo de los técnicos**. Estas herramientas deberán ser actualizadas también, especialmente el sistema de gestión de reservas que permite planificar el uso que los profesores hacen de las aulas y muestra información a los estudiantes acerca de la disponibilidad de las mismas. El gestor de reservas fue desarrollado por el propio laboratorio editando el código fuente de Drupal, el CMS sobre el que se construyó el sitio web. Esta es la razón por la que la web no puede ser simplemente actualizada: si se actualizase Drupal el gestor de reservas sería sobrescrito y se perdería.

Para resolver este problema, el laboratorio propuso diseñar un **nuevo sitio web usando CMSs y APIs más modernas** que permitieran la creación de un nuevo gestor de reservas que garantizara las futuras actualizaciones del sitio sin pérdida de datos y cuyo mantenimiento fuese más simple. Además, se pide que este nuevo gestor de reservas incluya funcionalidades adicionales que faciliten el trabajo de los técnicos de laboratorio. A lo largo de este documento se explican todas las decisiones de diseño tomadas para llevar a cabo este trabajo y desarrollar el proyecto.

1.2. Objetivos

El objetivo del proyecto es actualizar el sitio web del laboratorio y crear un nuevo sistema de reservas que incluya nuevas características y un diseño capaz de hacer que el trabajo de los técnicos del laboratorio para gestionar las reservas sea menos tedioso. Con la nueva aplicación se podrá:

- **Gestionar todas las aulas y sus reservas** con una nueva interfaz adaptada a móviles y tablets basada en HTML5 y Javascript.
- **Comprobar el software y las máquinas virtuales** necesarias en cada asignatura de forma automática.

- **Los profesores podrán solicitar reservas** para sus asignaturas desde la aplicación, que se encargará automáticamente de notificarla a los técnicos y permitir que estos la confirmen con un solo clic.
- **Generar estadísticas** sobre el uso de las clases y descargarlas en un documento de Microsoft Excel.

Además, la aplicación debe ser actualizable y garantizar la máxima compatibilidad con todas las nuevas versiones que sean lanzadas para el CMS elegido como base. La aplicación debe ser diseñada para durar.

1.3. Resto del documento

Este documento es una memoria que contiene la información sobre el desarrollo de *Gestor de Reservas*. Está estructurado de la siguiente manera:

1. **Introducción.** Incluye las motivaciones y objetivos que se esperan cumplir durante el desarrollo de la aplicación.
2. **Estado del arte.** Explica la actual situación del gestor de reservas del laboratorio e incluye información sobre las actuales soluciones comerciales que podrían ser usadas en este proyecto.
3. **Análisis.** Explora las alternativas definidas en el estado del arte y busca la mejor de entre ellas para usarla en el proyecto. Establece el alcance del proyecto con el cliente y comienza las primeras etapas de diseño.
4. **Diseño.** El diseño técnico de la aplicación donde se explica cada decisión tomada durante el proceso de desarrollo.
5. **Implementación e implantación.** En esta parte encontraremos información sobre el proceso de instalación y establecemos un plan de pruebas que garantice el correcto funcionamiento y la calidad del proyecto.
6. **Evaluación.** Define la batería de pruebas que debe pasar el sistema antes de alcanzar la versión 1.0.
7. **Planificación y presupuesto.** Incluye la planificación del proyecto y los costes (en dinero) asociados al desarrollo.
8. **Conclusiones y trabajos futuros.** Impresiones una vez terminado el desarrollo, también se proponen posibles nuevas ideas que podrían incluirse en el proyecto en un futuro.

En este documento también se encuentran referencias a la bibliografía usada en el proyecto así como las páginas webs visitadas para obtener información. En los apéndices encontrará el **manual de administración** usado para formar al personal del laboratorio en el uso de la aplicación.

2. Estado de la cuestión

2.1. Estado del arte

La solicitud del cliente consiste en la **renovación y actualización del sitio web del Laboratorio del Departamento de Informática**, junto con el diseño técnico y posterior implementación de una aplicación para la gestión de reservas de las aulas docentes de dicho departamento que sea soportado por ésta web. La aplicación debe no solo permitir realizar las gestiones de administración de estas reservas, sino también ofrecer la oportunidad a los profesores de realizar de forma simple sus reservas y a los alumnos de comprobar las aulas del laboratorio dónde se impartirán las clases de las asignaturas cursadas.

El sistema diseñado hace especial **énfasis en la sencillez** a la hora de la realización de las reservas para facilitar así un proceso que puede llegar a ser muy complejo, y ofrece a los profesores y alumnos información visual para conocer de un solo vistazo la ocupación de las aulas en cada momento del día.

Con lo expuesto anteriormente, es fácil situar como **principales interesados** en el sistema a los técnicos del Laboratorio del Departamento de Informática, así como los profesores y alumnos que hacen uso de las instalaciones de dicho departamento, además del propio LDI que actúa como cliente y del autor y desarrollador de este proyecto:

- **Laboratorio del Departamento de Informática:** su interés radica en que la aplicación satisfaga todas sus necesidades, que cumpla con todos sus requisitos, que sea lo más barata posible y que esté acabada cuanto antes.
- **Técnicos del LDI:** tiene como principal interés que el sistema facilite el proceso de reservas de aulas y les permita realizar los diferentes trabajos asociados de forma más cómoda y eficiente.
- **Profesores:** la aplicación debe simplificar de forma notable la forma en que los profesores solicitan las reservas de las aulas del departamento. Estos usuarios esperan que el sistema les guíe durante el proceso y les proporcione información en cada momento.
- **Alumnos:** los alumnos utilizarán la aplicación principalmente con el objetivo de conocer en qué aulas se imparten sus clases además de comprobar la disponibilidad de las aulas destinadas al trabajo libre.

- **Autor del proyecto:** el principal interés del autor de este proyecto será el de proporcionar un sistema con la máxima calidad, de manera eficiente y cumpliendo con los plazos estipulados en el contrato. Cumplir todos los requisitos del LDI, con el fin de que este pueda recomendarle en un futuro a otros posibles clientes.

El estudio de la situación actual viene dividido en tres partes bien diferenciadas en tanto que el propio proyecto consta de esas tres partes: la actualización de la página web del laboratorio, la elección del entorno de la máquina servidora que proveerá los servicios de dicha web y el diseño y desarrollo de una aplicación para la gestión de reservas de las aulas docentes del LDI.

2.2. Content Management System

Dado que el mantenimiento de un sitio web de estas características puede ser costoso y que el laboratorio no cuenta con personal dedicado exclusivamente a esta tarea, uno de los puntos acordados desde el primer momento con el cliente fue el **uso de un Content Management System o CMS**, que permitiera delegar las principales tareas de seguridad y corrección de errores a los creadores de este sistema. Así pues, debemos analizar los principales CMS existentes en el mercado y ver en qué medida prestan las funcionalidades necesarias para el desarrollo de nuestro sistema:



Drupal

Drupal es un CMS libre, modular, multipropósito y muy configurable que permite publicar artículos, imágenes, archivos y otros servicios añadidos como foros, encuestas, votaciones, blogs y administración de usuarios y permisos. Funciona bajo PHP y soporta bases de datos MySQL y SQLite.

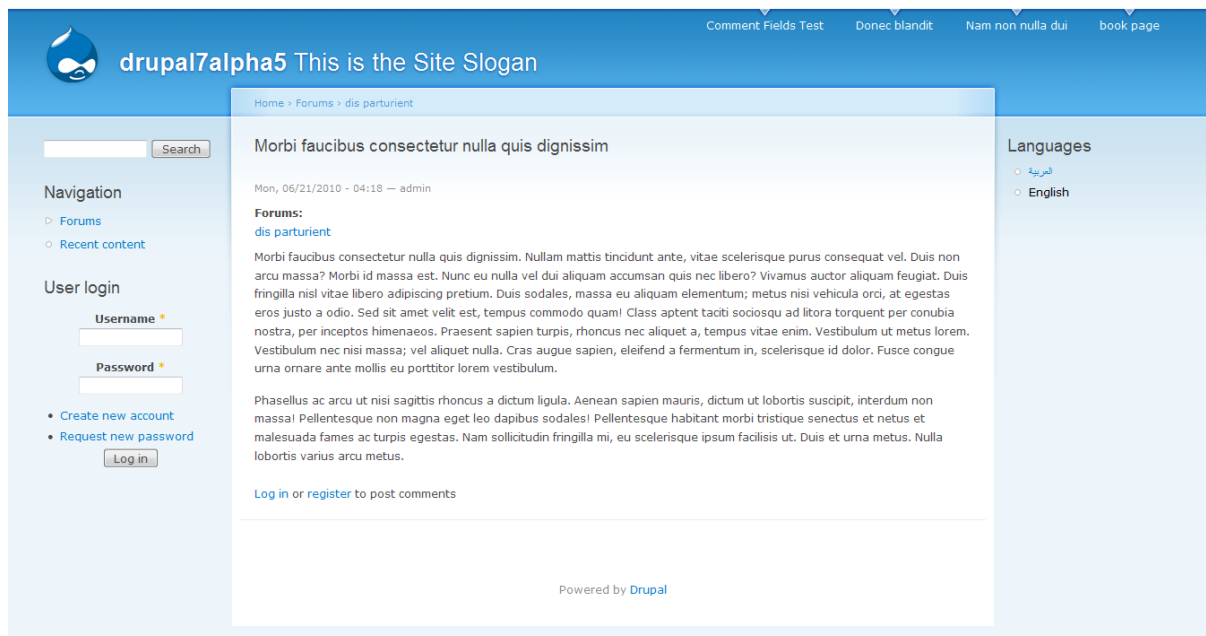


Figura 4: Página de inicio de Drupal

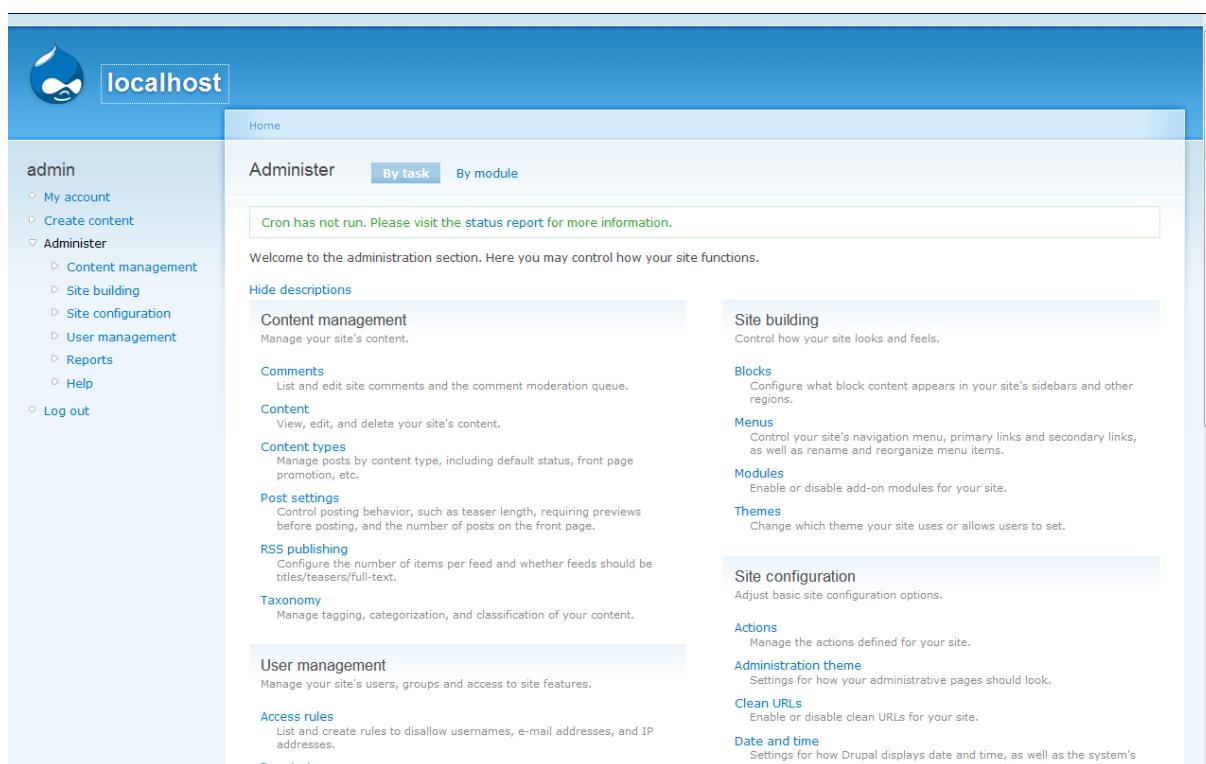


Figura 5: Panel de administración de Drupal

WordPress



WordPress es un CMS enfocado a la creación de blogs. Ha sido desarrollado en PHP para entornos que ejecuten MySQL y Apache, bajo licencia GPL. Se trata probablemente del CMS más popular en la actualidad y su uso sigue creciendo cada día gracias a su facilidad de uso, su funcionalidad y a la enorme comunidad de desarrolladores y diseñadores que crean contenido para él.

Twenty Fifteen

The WordPress default theme for 2015.

About Twenty Fifteen

Read me.

Readability

Twenty Fifteen supports optional menu description.

Image Alignment and Styles

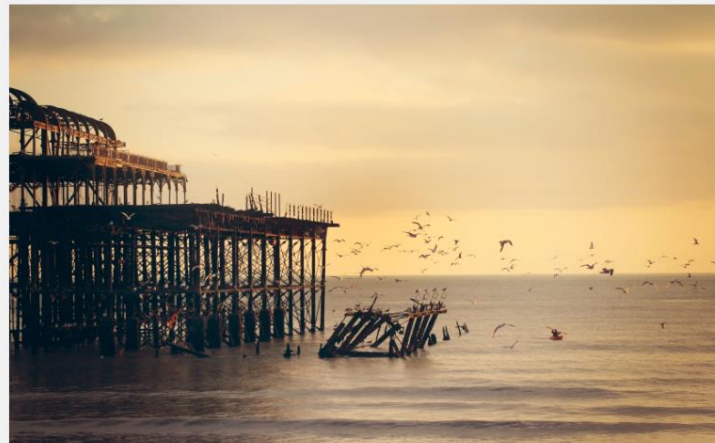
Check out image alignment style here.

HTML Elements

Check out styles for some HTML elements.

A Parent Page

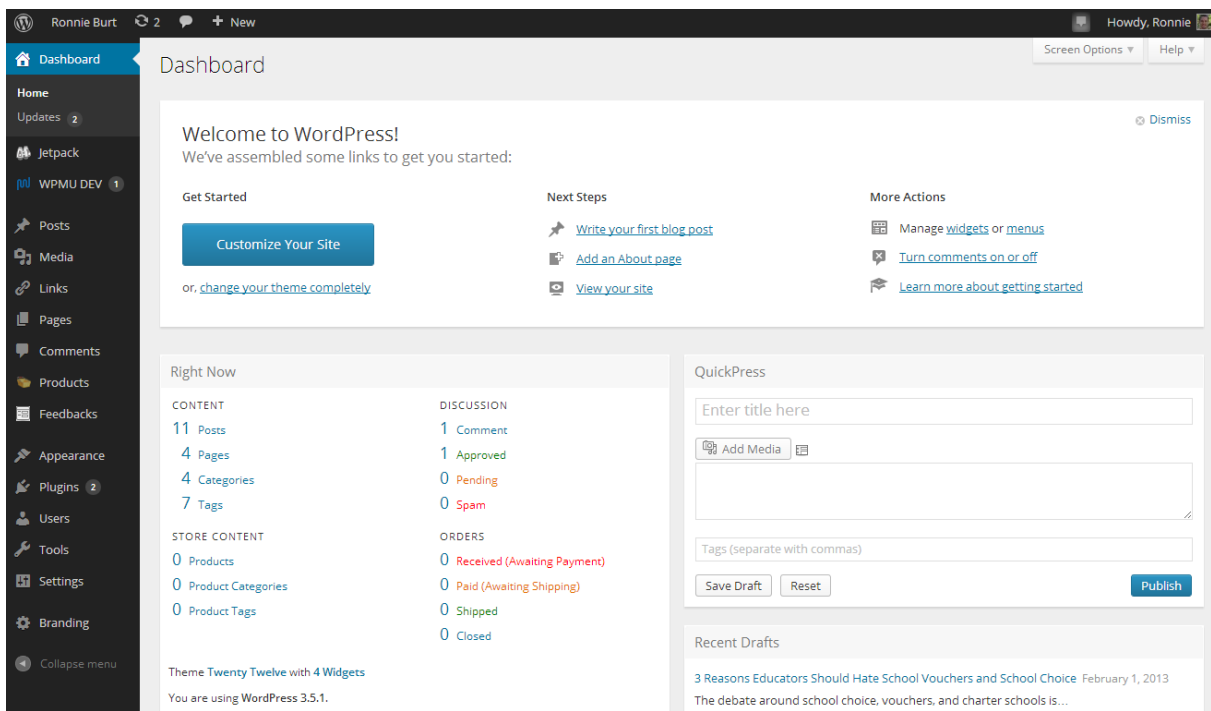
This is a dropdown demo.



Sticky Post with a Featured Image

Follow

Figura 6: Página de inicio de WordPress



The screenshot shows the WordPress Dashboard for a user named Ronnie Burt. The interface includes a sidebar with navigation links for Home, Updates, Jetpack, WPMU DEV, Posts, Media, Links, Pages, Comments, Products, Feedbacks, Appearance, Plugins, Users, Tools, Settings, Branding, and a Collapse menu button. The main content area is titled 'Dashboard' and features a 'Welcome to WordPress!' message with links to 'Get Started' (Customize Your Site, change your theme completely) and 'Next Steps' (Write your first blog post, Add an About page, View your site). There are also 'More Actions' links for managing widgets, turning comments on/off, and learning more about getting started. Below this, there are sections for 'Right Now' (Content: 11 Posts, 4 Pages, 4 Categories, 7 Tags; Store Content: 0 Products, 0 Product Categories, 0 Product Tags; Discussion: 1 Comment, 1 Approved, 0 Pending, 0 Spam; Orders: 0 Received, 0 Paid, 0 Shipped, 0 Closed) and 'QuickPress' (a form to create a new post with fields for title, media, tags, and buttons for Save Draft, Reset, and Publish). At the bottom, there is a 'Recent Drafts' section showing a draft titled '3 Reasons Educators Should Hate School Vouchers and School Choice' from February 1, 2013.

Figura 7: Página de administración de WordPress

Análisis, diseño e implantación de una aplicación para la gestión de reservas para las aulas docentes del departamento de informática



Joomla

Joomla es un sistema de gestión de contenidos que permite desarrollar sitios web dinámicos e interactivos. Permite crear, modificar o eliminar contenido de un sitio web de manera sencilla a través de un panel de administración. Es un software de código abierto, desarrollado en PHP y con licencia GPL.

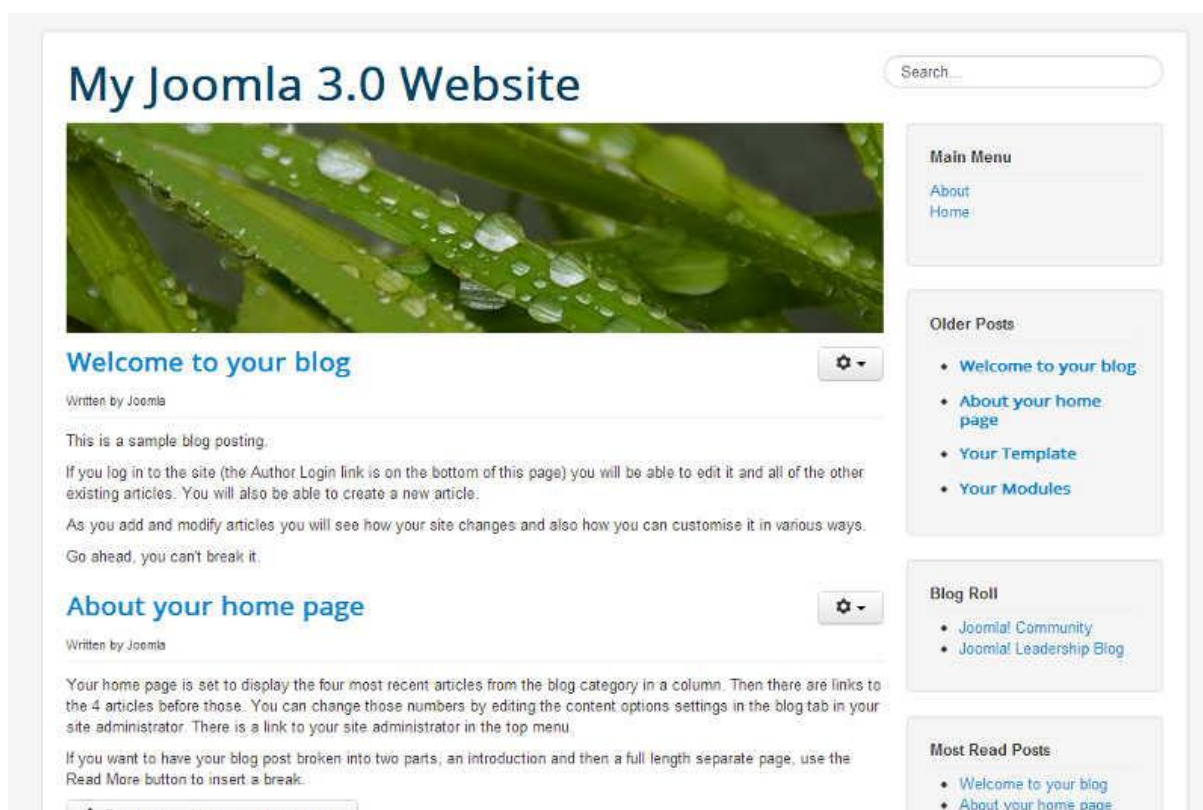


Figura 8: Página de inicio de Joomla

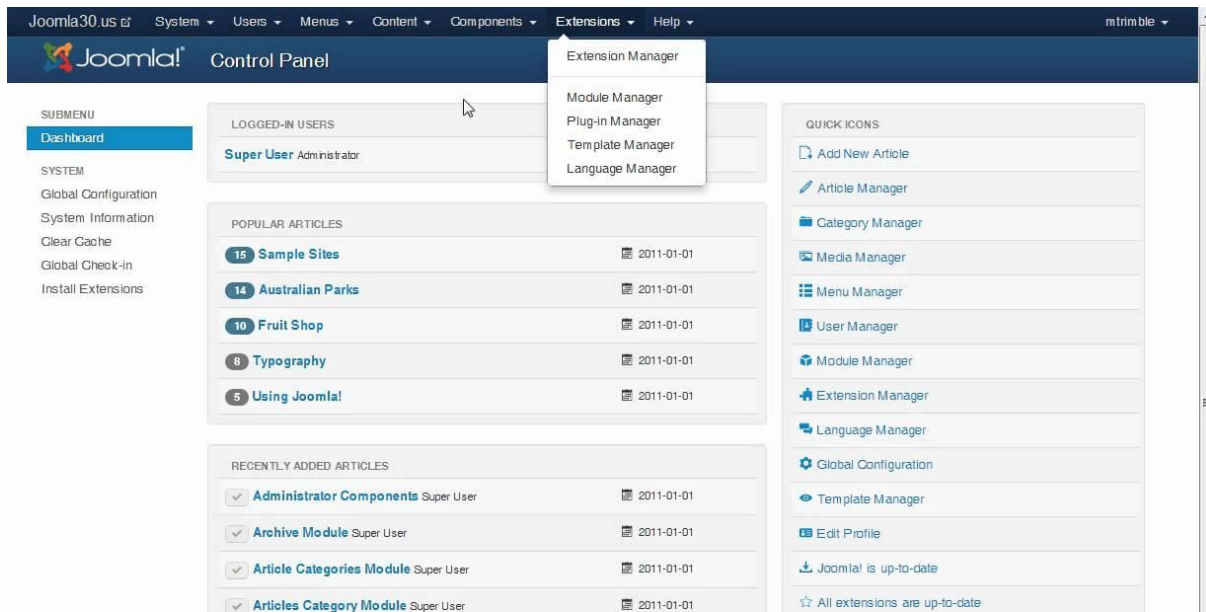


Figura 9: Página de administración de Joomla

Aplicaciones similares al gestor de reservas

Otra de las cuestiones que sin duda debemos evaluar de cara a la implantación de la aplicación para la gestión de reservas para las aulas docentes del departamento de informática es la **existencia previa de soluciones ya desarrolladas** y que puedan suplir todas nuestras necesidades, o al menos pudiesen servir como base del desarrollo de nuestra propia implementación. Dado que es seguro que la nueva página web del LDI funcionará bajo alguno de los CMS estudiados en el apartado anterior, la búsqueda de las alternativas a la aplicación se realizará en este marco de operación:

Módulos

Drupal no ofrece por defecto ninguna funcionalidad similar a la que el cliente necesita y que debemos ofrecerle. Sin embargo este CMS puede ser ampliado con nuevas funcionalidades gracias a paquetes de software desarrollados por terceros llamados **módulos**. Tras realizar una búsqueda exhaustiva encontramos los siguientes candidatos posibles para realizar las funciones necesarias:

- **Scheduler.** Este módulo, desarrollado por Eric Schaefer permite la configuración de Drupal para mostrar u ocultar ciertas publicaciones en función de la fecha y la hora lo que podría permitir su uso como un gestor de reservas de aulas que es lo que nosotros necesitamos.

- **Aplicación desarrollada por el LDI.** El departamento desarrolló en su día una aplicación que es la que gobierna actualmente el sistema de reservas de la página Web. Sin embargo este desarrollo no se hizo siguiendo el estándar de módulos de Drupal, sino que se realizó una modificación del código fuente del CMS sobre una versión bastante antigua del mismo. Así, una posibilidad sería adaptar esta aplicación como un módulo de Drupal aprovechando el trabajo anterior.

Plugins

WordPress tampoco implementa por defecto una funcionalidad que permita la gestión de aulas o la planificación de diferentes actividades en un marco de tiempo. Por tanto debemos de nuevo recurrir a la búsqueda de **plugins** (nombre que reciben los módulos en WordPress) desarrollados por terceros para intentar encontrar una solución ya implementada que resuelva nuestro problema y facilite la realización del proyecto:

- **The Events Calendar.** Este plugin de WordPress permite la planificación de diferentes eventos que se exponen en un calendario mensual y que pueden ser vistos por cualquier usuario que visite la web. Cuenta también con una vista de agenda que muestra los eventos más próximos a la fecha actual.
- **Weekly Class Schedule.** Este planificador ofrece distintas vistas por mes y semana o en forma de lista, asignación de reservas por aulas personalizables y una interfaz intuitiva y agradable.

Extensiones

Para desempeñar la función de gestor de reservas, Joomla también necesitará el uso de **extensiones** (nombre que reciben en este CMS los llamados módulos en Drupal y plugins en Wordpress). De entre los más populares, destacamos dos de ellos:

- **Scheduler.** Se trata de una completa y potente extensión que contiene multitud de características. Permite la vista por día, semana y mes. Las opciones de configuración son extremadamente completas y muy flexibles para adaptar la aplicación a diferentes ámbitos.
- **Schedule School.** Se trata de un sencillo calendario que muestra las reservas o eventos planificados dentro de una vista semanal de 5 días (de lunes a viernes).

2.4. APIs y otras herramientas

Podría darse el caso de que ninguno de los módulos que extienden el funcionamiento de cada uno de los CMS analizados sea capaz de cumplir con las características y requisitos que necesitamos para el sistema gestor de reservas. Es por eso que podría ser conveniente analizar algunas **herramientas y APIs** que pudieran servirnos para facilitar la tarea en caso de que fuese necesario desarrollar nuestra propia aplicación de gestión de reservas. El propio LDI, actuando como cliente de la aplicación propone al desarrollador el uso de las siguientes herramientas:

- **Jtable.** Se trata de un plugin de JQuery desarrollado en AJAX que permite la creación de tablas personalizadas en las que mostrar información simplificando enormemente el proceso de desarrollo. Dado que la aplicación de gestión de reservas debe mostrar gran número de información, esta herramienta puede ser de mucha ayuda durante el desarrollo.
- **Generación de documentos Excel (.xls) con PHP.** El LDI propone también utilizar las funciones de PHP para el desarrollo de una pequeña herramienta que permita la generación de documentos de Microsoft Excel.

2.5. Entorno del servidor

En último lugar, es importante realizar un estudio de las posibles **alternativas de entorno tecnológico** más adecuadas para el despliegue de la nueva página web del LDI y de la aplicación para la gestión de reservas de las aulas docentes.

El servidor supone la parte del hardware más importante de nuestro sistema y es el motor de nuestra aplicación. Debido a las necesidades del laboratorio, será requisito indispensable que nuestro sistema sea capaz de funcionar en la **misma máquina en la que lo hacía la versión original** de la página del LDI. Esta máquina tiene las siguientes especificaciones:

- RAM: 8GB
- MEMORIA: 512GB HDD
- Sistema Operativo: Debian Jessie

No obstante, no podemos utilizar toda la capacidad de la máquina para nuestra aplicación, pues dentro de este servidor conviven varias aplicaciones

más como la página del Departamento de Informática. Para esta convivencia, el departamento recurre al uso de varias máquinas virtuales con XEN dentro del servidor que garantizan la seguridad y el buen funcionamiento de cada una de estas aplicaciones de forma independiente. La máquina virtual actual de la página web del LDI cuenta con **las siguientes especificaciones técnicas**:

- RAM: 2GB con 512MB de swap
- MEMORIA: 20GB
- Sistema Operativo: Debian Jessie

Sería recomendable pues, tratar de garantizar que la nueva web del LDI sea capaz de funcionar de forma correcta con estas mismas especificaciones y en una máquina virtual similar. Teniendo en cuenta estos requisitos, podemos establecer **una serie de alternativas** para el entorno tecnológico de nuestra aplicación:

La primera alternativa consiste en el uso de la última versión de Ubuntu con soporte extendido junto con un servidor apache y una base de datos no relacional.

Alternativa 1	
Sistema Operativo	Ubuntu 14.04 LTS
Servidor Web	Apache HTTP Server
Sistema Gestor de Bases de Datos	MongoDB

Tabla 2: Alternativa de servidores 1

La segunda alternativa es el uso de otra de las distribuciones Linux más populares, con una configuración más conservadora para la persistencia de datos.

Alternativa 2	
Sistema Operativo	Debian Jessie
Servidor Web	Apache HTTP Server
Sistema Gestor de Bases de Datos	MySQL

Tabla 3: Alternativa de servidores 2

Para la última opción se propone como sistema operativo, servidor web y sistema gestor de bases de datos el software propietario de Microsoft.

Alternativa 3	
Sistema Operativo	Windows Server 2012 R2
Servidor Web	Internet Information Services 8 (IIS 8)
Sistema Gestor de Bases de Datos	Microsoft SQL Server

Tabla 4: Alternativa de servidores 3

3. Análisis

3.1. Introducción

La aplicación *Gestor de Reservas* es un producto innovador pensado para facilitar la **gestión de las reservas de las aulas del Laboratorio del Departamento de Informática**, permitiendo una liberación de trabajo y mejora del entorno laboral a los técnicos del laboratorio. *Gestor de Reservas* cuenta con una interfaz de usuario en castellano que permite la gestión de las reservas a los técnicos del laboratorio al mismo tiempo que ofrece una manera sencilla de que los alumnos y profesores de la universidad puedan consultar la ocupación de las aulas del LDI.

El entorno tecnológico de la aplicación está compuesto por los siguientes elementos:

- **Panel de administración:** desde el panel de administración los técnicos del laboratorio podrán gestionar de forma eficiente las reservas de las aulas del departamento.
- **Panel de usuario:** desde el panel de usuario los profesores podrán solicitar la reserva de las aulas del departamento.
- **Calendario:** una vista en forma de calendario con la ocupación de las reservas en diferentes periodos.
- **Servidor de apoyo:** en este grupo se incluyen el servidor de base de datos y otros servidores que realizan tareas de apoyo y soporte del sistema.
- El **CMS** sobre el que se construye la aplicación.

Tanto para el diseño de las distintas funcionalidades del sistema así como para la realización de la documentación necesaria para el seguimiento del proceso de desarrollo, se hará uso de **varios estándares** usados habitualmente en la ingeniería del software:

- **Guía para la documentación de la memoria del TFG:** para redactar esta memoria se utilizará el índice de contenidos facilitado por la Universidad Carlos III con alguna leve modificación en la estructura en caso de que fuese necesario.
- **Estándar IEEE 730 – 2002**, IEEE standard for Software Quality Assurance Plans: se seguirá dicho estándar para la redacción de los requisitos del sistema.
- **Estándar ISO/IEC 19501:** se seguirá dicho estándar para la realización de los diagramas.

- **Estándar ISO/IEC 29119:** se seguirá dicho estándar para la realización de las pruebas.

Para el análisis será necesario también detallar los **stakeholders** que soportarán el desarrollo y que serán tenidos en cuenta a la hora evaluar la adecuación del producto final:

- **Cliente:** el Laboratorio del Departamento de Informática será el responsable de validar el análisis realizado en este documento y de ofrecer soporte durante toda la duración del proyecto. Además su interés radica en que la aplicación satisfaga sus necesidades y que cumpla con todos su requisitos para garantizar el éxito del proyecto.
- **Usuarios:** usuarios finales que utilizarán la aplicación. Son los técnicos del LDI y también los profesores y alumnos de la universidad que hacen uso de las aulas con fines docentes. Serán evaluadores de las funcionalidades del producto, aconsejando sobre qué mejorar o qué funcionalidades añadir o eliminar.
- **Desarrollador:** el responsable de la creación del proyecto que busca superar el TFG para obtener el título de Graduado en Ingeniería Informática.

3.2. Valoración de las alternativas

El primer paso del Análisis consistirá en la realización de un **estudio en profundidad de las posibles implementaciones propuestas** en *2. Estado de la Cuestión*, a fin de conocer mejor los puntos fuertes y débiles de cada una de las alternativas que tenemos disponibles para cada uno de los puntos que han sido estudiados.

Content Management System

Después de haber descrito cada uno de los CMS de manera individual, es el momento de **comparar cada una de las soluciones** y sus características y atributos con el fin de destacar los puntos fuertes y débiles de cada una de ellas, de manera que con nuestra elección aprovechemos toda la ventaja disponible para el diseño de nuestra aplicación.

Las características o **atributos más relevantes** para el diseño de una aplicación web como la que nosotros necesitamos y qué soporte nuestro sistema de gestión de reservas son descritos a continuación:

- **Diseño:** el diseño es una parte esencial para cualquier aplicación que requiera de una interacción directa con el usuario, ya que no únicamente determina la vistosidad, sino también la diferencia entre la facilidad de uso o una mala experiencia para el usuario. En definitiva, la aplicación web debe seguir las líneas de diseño vigentes en la actualidad, con un diseño *responsive* para ser visto de forma cómoda en dispositivos móviles, minimalista, cómodo y atractivo a la vista.
- **Rendimiento:** permite que el funcionamiento de la aplicación sea fluido, consumiendo el mínimo de recursos, permitiendo además una mayor cantidad de usuarios simultáneos y proporcionando la mejor experiencia de uso posible gracias a una navegación eficiente.
- **Funcionalidad:** es un factor clave a tener en cuenta, no solo para la implementación de nuestro sistema sino pensando en actualizaciones futuras. La plataforma elegida debe ser capaz de adaptarse a nuevas funcionalidades además de aquellas para las que se pensó inicialmente.
- **Facilidad de uso.** La administración de sitios webs puede ser compleja si se requiere la actualización habitual de contenido. Por eso debe primar la sencillez y la eficiencia a la hora de modificar o añadir contenido a la aplicación.
- **Mantenimiento.** Otra tarea vital de un sistema basado en web es el mantenimiento del mismo. Es vital que el CMS elegido mantenga una política de actualizaciones rápida que permita solventar eventuales bugs o problemas de seguridad que fuesen descubiertos y pudiesen comprometer el rendimiento del sistema. Es importante también que estas actualizaciones se realicen de forma sencilla y transparente, y no causen problemas de uso en complementos que puedan estar en funcionamiento.
- **Existencia de contenido adicional.** Como hemos visto en el apartado destinado a cada opción, debemos evaluar si alguna de las propuestas de módulos existentes puede ayudarnos o servirnos como base para el desempeño de las tareas de planificación de las reservas en el Laboratorio.
- **Creación de contenido adicional.** Por último y dado que el propósito final es la creación de una aplicación para la gestión de reservas para las aulas docentes del departamento de informática que funcione bajo el CMS escogido es primordial que dicho gestor de contenidos permita la creación de contenido adicional para el mismo de una forma simple y bien documentada.

Como cabría esperar, **ninguna de las soluciones existente puede cumplir al completo con todas las características** deseables para nuestro sistema. A continuación se expone una tabla comparativa para proporcionar una descripción más visual de los puntos fuertes y débiles de cada alternativa y la medida en que satisfacen nuestras necesidades (Alto, Medio, Bajo):

	Drupal	Wordpress	Joomla
Diseño	Medio	Alto	Alto
Rendimiento	Alto	Medio	Medio
Funcionalidad	Bajo	Medio	Alto
Facilidad de uso	Bajo	Alto	Medio
Mantenimiento	Medio	Alto	Medio
Contenido	Bajo	Bajo	Bajo
Crear contenido	Bajo	Alto	Medio

Tabla 5: Comparativa de CMS

WordPress parece ser la opción con mejores resultados en nuestra comparativa, aunque en verdad no obtiene los mejores resultados ni en rendimiento ni en funcionalidad. No obstante su puntuación en el resto de características hace que sea finalmente la plataforma elegida sobre la que desarrollar nuestra aplicación.

Plugins

Con la elección del CMS realizada, ahora debemos estudiar si es posible el **uso del software desarrollado por terceros** para WordPress detallado en el apartado 2.3. *Aplicaciones Similares al Gestor de Reservas*. Estudiaremos las dos alternativas candidatas, The Events Calendar y Weekly Class Schedule, atendiendo a los siguientes criterios preestablecidos que debe cumplir el gestor de las aulas docentes del LDI más básico:

- **Configuración de aulas docentes.** Las aulas docentes del departamento no son fijas. En 2014 algunas de ellas dejaron de estar disponibles y se añadieron otras nuevas, algo que puede que se repita en el futuro. Por lo tanto, el plugin elegido debe ser capaz de modificar fácilmente las aulas.
- **Vista diaria.** Los alumnos deben poder ver cada día las reservas de todas las aulas de un solo vistazo, esto es, ofreciendo una vista diaria donde se muestren todas las reservas de todas las aulas de forma fácil de entender.

- **Vista por aula.** También debe ser posible ver las reservas de toda la semana para un aula determinada, a modo de facilitar la gestión de las reservas y la planificación de las clases.

Analicemos ahora si nuestras alternativas cumplen o no con los criterios mínimos establecidos que debe tener la aplicación de gestión de las aulas docentes del LDI:

	The Event Calendar	Weekly Class Schedule
Configurar Aulas	No	Sí
Vista diaria	No	No
Vista por aulas	Sí	Sí

Tabla 6: Comparativa de Plugins

Puesto que como hemos visto **no existe ningún plugin para WordPress que cumpla nuestras necesidades mínimas**, concluimos que deberemos desarrollar una nueva extensión para este objetivo. La plataforma condiciona las tecnologías de desarrollo que deberemos utilizar para la implementación de nuestra aplicación para la gestión de reservas para las aulas docentes del departamento de informática. Así, *WordPress* permite la creación de *plugins* (es el nombre que da a sus complementos) escritos en lenguaje PHP. Además, permite el uso de otras tecnologías webs populares como HTML, CSS y Javascript. En cuanto a la persistencia de datos, *WordPress* provee de una interfaz para hacer uso de su base de datos *MySQL*. Este será, por lo tanto, el entorno sobre el que desarrollaremos la aplicación.

Selección del entorno operacional

Aún queda por resolver **la cuestión del entorno tecnológico** que debe emplearse para la ejecución de la aplicación web bajo *WordPress* de la forma más eficiente. Pese a que se estudió la posibilidad de realizar un estudio de las posibles alternativas para una posterior valoración de las mismas, por cuestiones económicas dentro del Laboratorio del Departamento de Informática debieron establecerse unas limitaciones muy estrictas en cuanto a los requisitos de hardware y software bajo los que correrá la aplicación.

Estos requisitos vienen dados por la **imposibilidad de renovar el equipo servidor encargado de alojar la página web del LDI**, debido principalmente a cuestiones de presupuesto. Así, se convierte en

requisito indispensable que la aplicación final pueda ejecutarse con normalidad en un servidor con los mismos requisitos de hardware y software con los que cuenta el que aloja la actual página web. Estos requisitos son recogidos en la siguiente tabla:

Requisitos de ejecución	
Sistema Operativo	Debian Jessie
Servidor Web	Apache HTTP Server
Sistema Gestor de Bases de Datos	MySQL
Memoria	2GB RAM, 20GB

Tabla 7: Requisitos de ejecución

Estos requisitos más que suficientes para la correcta ejecución de WordPress, por lo que nuestra misión será **garantizar que el plugin diseñado para él no suponga una excesiva carga** que impida la correcta ejecución del conjunto del sistema.

3.3. Definición de los requisitos del sistema

Una vez definido el entorno de trabajo y la plataforma sobre la que se construirá la aplicación, definiremos los primeros requisitos que van a tenerse en cuenta a la hora de realizar el diseño. Cada uno de estos requisitos va a ir definido por una tabla similar a la siguiente:

IDENTIFICADOR	
PRIORIDAD: <input type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	FUENTE: <input type="checkbox"/> CLIENTE <input type="checkbox"/> DESARROLLADOR
NECESIDAD: <input type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
CLARIDAD: <input type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
ESTABILIDAD:	
DESCRIPCIÓN:	

Tabla 8: Ejemplo de tabla de requisitos

Donde cada uno de los campos de la tabla será rellenado de la siguiente forma:

- Identificador: sirve para distinguir un requisito de otro inequívocamente. Cada identificador seguirá la nomenclatura UX-Ynnn, donde:
 - U: Requisito de usuario.
 - X: Tipo de requisito, pudiendo distinguir:
 - R: rendimiento.
 - K: restricción.

- S: seguridad.
 - I: interfaz.
 - Y: Indica que el tipo de requisito:
 - C: es un requisito de capacidad (funcional).
 - R: es un requisito de restricción (no funcional).
 - nnn: indica el número de requisito con un valor comprendido entre 1 y 999.
- Prioridad: informa del grado de prioridad con el que debe ser resuelto un requisito. Podrá tener los siguientes valores:
 - Alta: el requisito es de carácter prioritario.
 - Media: el requisito tiene una prioridad media.
 - Baja: el requisito es de baja prioridad.
- Fuente: sobre el origen del requisito:
 - Cliente: requisito solicitado por los técnicos del Laboratorio del Departamento de Informática.
 - Desarrollador: requisito impuesto por el desarrollador.
- Necesidad: indica el grado de necesidad de incorporar el requisito en el sistema. Puede tomar los siguientes valores:
 - Esencial: el requisito debe estar contemplado necesariamente en el sistema.
 - Deseable: el requisito debería incluirse en el sistema siempre que no cause errores o conflictos con otros requisitos.
 - Opcional: el requisito puede o no estar reflejado en el producto final.
- Claridad: sobre si el requisito esta expresado correctamente y de forma sencilla:
 - Alta: el requisito solo tiene una interpretación.
 - Media: el requisito está bien definido y es improbable que surjan dudas.
 - Baja: el requisito puede tener varias interpretaciones dependiendo del contexto.
- Verificabilidad: este campo indica la posibilidad de comprobar que el requisito se haya incorporado al sistema:
 - Alta: puede comprobarse de manera sencilla.
 - Media: comprobar si el requisito se ha incorporado puede suponer algo de trabajo.
 - Baja: es prácticamente imposible comprobar si el requisito se cumple en el producto final.
- Estabilidad: define la posibilidad de que un requisito no se modifique durante el desarrollo del proyecto.
- Descripción: en este campo se incluye la definición del requisito.

Requisitos de Capacidad

UR-C001	
PRIORIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	FUENTE: <input checked="" type="checkbox"/> CLIENTE <input type="checkbox"/> DESARROLLADOR
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
ESTABILIDAD:	Durante toda la vida del sistema
DESCRIPCIÓN:	Los profesores pueden reservas aulas desde la aplicación

Tabla 9: Requisito UR-C001

UR-C002	
PRIORIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	FUENTE: <input checked="" type="checkbox"/> CLIENTE <input type="checkbox"/> DESARROLLADOR
NECESIDAD: <input type="checkbox"/> ESENCIAL <input checked="" type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
ESTABILIDAD:	Durante toda la vida del sistema
DESCRIPCIÓN:	Rellenar un formulario para solicitar la reserva.

Tabla 10: Requisito UR-C002

UR-C003	
PRIORIDAD: <input type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input checked="" type="checkbox"/> BAJA	FUENTE: <input checked="" type="checkbox"/> CLIENTE <input type="checkbox"/> DESARROLLADOR
NECESIDAD: <input type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input checked="" type="checkbox"/> OPCIONAL	
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
ESTABILIDAD:	Durante toda la vida del sistema
DESCRIPCIÓN:	Los profesores pueden renovar las reservas realizadas en cursos anteriores.

Tabla 11: Requisito UR-C003

UR-C004	
PRIORIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	FUENTE: <input checked="" type="checkbox"/> CLIENTE <input type="checkbox"/> DESARROLLADOR
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
ESTABILIDAD:	Durante toda la vida del sistema
DESCRIPCIÓN:	Los usuarios administradores pueden añadir, editar y modificar todas las reservas almacenadas en el sistema.

Tabla 12: Requisito UR-C004

UR-C005	
PRIORIDAD: <input type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input checked="" type="checkbox"/> BAJA	FUENTE: <input checked="" type="checkbox"/> CLIENTE <input type="checkbox"/> DESARROLLADOR
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
ESTABILIDAD:	Durante toda la vida del sistema
DESCRIPCIÓN:	Los administradores pueden aceptar o rechazar las reservas de los profesores.

Tabla 13: Requisito UR-C005

UR-C006	
PRIORIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	FUENTE: <input checked="" type="checkbox"/> CLIENTE <input type="checkbox"/> DESARROLLADOR
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
ESTABILIDAD:	Durante toda la vida del sistema
DESCRIPCIÓN:	Los usuarios pueden ver las reservas realizadas.

Tabla 14: Requisito UR-C006

UR-C007	
PRIORIDAD: <input type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input checked="" type="checkbox"/> BAJA	FUENTE: <input checked="" type="checkbox"/> CLIENTE <input type="checkbox"/> DESARROLLADOR
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
ESTABILIDAD:	Durante toda la vida del sistema
DESCRIPCIÓN:	Los administradores pueden indicar las aulas disponibles para reservar.

Tabla 15: Requisito UR-C007

UR-C008	
PRIORIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	FUENTE: <input checked="" type="checkbox"/> CLIENTE <input type="checkbox"/> DESARROLLADOR
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
ESTABILIDAD:	Durante toda la vida del sistema
DESCRIPCIÓN:	Los administradores pueden indicar las asignaturas que pueden realizar reservas.

Tabla 16: Requisito UR-C008

UR-C009	
PRIORIDAD: <input type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input checked="" type="checkbox"/> BAJA	FUENTE: <input checked="" type="checkbox"/> CLIENTE <input type="checkbox"/> DESARROLLADOR
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
ESTABILIDAD:	Durante toda la vida del sistema
DESCRIPCIÓN:	Los administradores pueden indicar los grados que pueden realizar reservas.

Tabla 17: Requisito UR-C009

UR-C010	
PRIORIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	FUENTE: <input checked="" type="checkbox"/> CLIENTE <input type="checkbox"/> DESARROLLADOR
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
ESTABILIDAD:	Durante toda la vida del sistema
DESCRIPCIÓN:	Los administradores pueden indicar el software instalado en las aulas.

Tabla 18: Requisito UR-C010

UR-C011	
PRIORIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	FUENTE: <input checked="" type="checkbox"/> CLIENTE <input type="checkbox"/> DESARROLLADOR
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
ESTABILIDAD:	Durante toda la vida del sistema
DESCRIPCIÓN:	Los administradores pueden indicar las máquinas virtuales disponibles en las aulas.

Tabla 19: Requisito UR-C011

UR-C012	
PRIORIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	FUENTE: <input type="checkbox"/> CLIENTE <input checked="" type="checkbox"/> DESARROLLADOR
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
CLARIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
ESTABILIDAD:	Durante toda la vida del sistema
DESCRIPCIÓN:	Los administradores pueden configurar los periodos de los cuatrimestres

Tabla 20: Requisito UR-C012

UR-C013	
PRIORIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	FUENTE: <input type="checkbox"/> CLIENTE <input checked="" type="checkbox"/> DESARROLLADOR
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
CLARIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
ESTABILIDAD:	Durante toda la vida del sistema
DESCRIPCIÓN:	Los administradores pueden cambiar el color con el que se muestran las reservas.

Tabla 21: Requisito UR-C013

UR-C014	
PRIORIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	FUENTE: <input type="checkbox"/> CLIENTE <input checked="" type="checkbox"/> DESARROLLADOR
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
CLARIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
ESTABILIDAD:	Durante toda la vida del sistema
DESCRIPCIÓN:	Los administradores pueden configurar la conexión con un servidor LDAP.

Tabla 22: Requisito UR-C014

UR-C015	
PRIORIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	FUENTE: <input type="checkbox"/> CLIENTE <input checked="" type="checkbox"/> DESARROLLADOR
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
ESTABILIDAD:	Durante toda la vida del sistema
DESCRIPCIÓN:	Permitir la inclusión del resto de características necesarias para la web del LDI.

Tabla 23: Requisito UR-C015

UI-C016	
PRIORIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	FUENTE: <input checked="" type="checkbox"/> CLIENTE <input type="checkbox"/> DESARROLLADOR
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
ESTABILIDAD:	Durante toda la vida del sistema
DESCRIPCIÓN:	Mostrar el logo del Laboratorio del Departamento de Informática.

Tabla 24: Requisito UI-C016

UI-C017	
PRIORIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	FUENTE: <input checked="" type="checkbox"/> CLIENTE <input type="checkbox"/> DESARROLLADOR
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
ESTABILIDAD:	Durante toda la vida del sistema
DESCRIPCIÓN:	Mostrar el calendario de reservas por día.

Tabla 25: Requisito UI-C017

UI-C018	
PRIORIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	FUENTE: <input checked="" type="checkbox"/> CLIENTE <input type="checkbox"/> DESARROLLADOR
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
ESTABILIDAD:	Durante toda la vida del sistema
DESCRIPCIÓN:	Mostrar el calendario de reservas por aula.

Tabla 26: Requisito UI-C018

UI-C019	
PRIORIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	FUENTE: <input checked="" type="checkbox"/> CLIENTE <input type="checkbox"/> DESARROLLADOR
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
ESTABILIDAD:	Durante toda la vida del sistema
DESCRIPCIÓN:	Mostrar la lista de reservas

Tabla 27: Requisito UI-C019

UI-C020	
PRIORIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	FUENTE: <input checked="" type="checkbox"/> CLIENTE <input type="checkbox"/> DESARROLLADOR
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
ESTABILIDAD:	Durante toda la vida del sistema
DESCRIPCIÓN:	Mostrar la lista de aulas

Tabla 28: Requisito UI-C020

UI-C021	
PRIORIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	FUENTE: <input checked="" type="checkbox"/> CLIENTE <input type="checkbox"/> DESARROLLADOR
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
ESTABILIDAD:	Durante toda la vida del sistema
DESCRIPCIÓN:	Mostrar la lista de asignaturas

Tabla 29: Requisito UI-C021

UI-C022	
PRIORIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	FUENTE: <input checked="" type="checkbox"/> CLIENTE <input type="checkbox"/> DESARROLLADOR
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
ESTABILIDAD:	Durante toda la vida del sistema
DESCRIPCIÓN:	Mostrar la lista de grados.

Tabla 30: Requisito UI-C022

UI-C023	
PRIORIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	FUENTE: <input checked="" type="checkbox"/> CLIENTE <input type="checkbox"/> DESARROLLADOR
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
ESTABILIDAD:	Durante toda la vida del sistema
DESCRIPCIÓN:	Mostrar la lista de software instalado.

Tabla 31: Requisito UI-C023

UI-C024	
PRIORIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	FUENTE: <input checked="" type="checkbox"/> CLIENTE <input type="checkbox"/> DESARROLLADOR
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
ESTABILIDAD:	Durante toda la vida del sistema
DESCRIPCIÓN:	Mostrar la lista de máquinas virtuales instaladas.

Tabla 32: Requisito UI-C024

UI-C025	
PRIORIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	FUENTE: <input checked="" type="checkbox"/> CLIENTE <input type="checkbox"/> DESARROLLADOR
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
ESTABILIDAD:	Durante toda la vida del sistema
DESCRIPCIÓN:	Mostrar informes de las reservas.

Tabla 33: Requisito UI-C025

Requisitos de Restricción

UK-R001	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	FUENTE: <input type="checkbox"/> CLIENTE <input checked="" type="checkbox"/> DESARROLLADOR
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
ESTABILIDAD:	Durante toda la vida del sistema
DESCRIPCIÓN:	Compatible con Wordpress a partir de la versión 3.5.

Tabla 34: Requisito UK-R001

UK-R002	
PRIORIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	FUENTE: <input checked="" type="checkbox"/> CLIENTE <input type="checkbox"/> DESARROLLADOR
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
CLARIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
ESTABILIDAD:	Durante toda la vida del sistema
DESCRIPCIÓN:	Compatible con navegadores web de escritorio y móvil.

Tabla 35: Requisito UK-R002

UR-R003	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	FUENTE: <input type="checkbox"/> CLIENTE <input checked="" type="checkbox"/> DESARROLLADOR
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
ESTABILIDAD:	Durante toda la vida del sistema
DESCRIPCIÓN:	El tiempo de carga/respuesta no debe superar en ningún caso los 5 segundos.

Tabla 36: Requisito UR-R003

US-R004	
PRIORIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	FUENTE: <input type="checkbox"/> CLIENTE <input checked="" type="checkbox"/> DESARROLLADOR
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
CLARIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
ESTABILIDAD:	Durante toda la vida del sistema
DESCRIPCIÓN:	Garantizar la seguridad de los datos

Tabla 37: Requisito US-R004

UR-R005	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	FUENTE: <input type="checkbox"/> CLIENTE <input checked="" type="checkbox"/> DESARROLLADOR
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
ESTABILIDAD:	Durante toda la vida del sistema
DESCRIPCIÓN:	Estar inactivo un máximo de 30 minutos al mes por motivos de mantenimiento.

Tabla 38: Requisito UR-R005

UI-R006	
PRIORIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	FUENTE: <input type="checkbox"/> CLIENTE <input checked="" type="checkbox"/> DESARROLLADOR
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
CLARIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
ESTABILIDAD:	Durante toda la vida del sistema
DESCRIPCIÓN:	Tener una interfaz clara y sencilla, para facilitar el uso a todo tipo de usuarios.

Tabla 39: Requisito UI-R006

UI-R007	
PRIORIDAD: <input type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input checked="" type="checkbox"/> BAJA	FUENTE: <input type="checkbox"/> CLIENTE <input checked="" type="checkbox"/> DESARROLLADOR
NECESIDAD: <input type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input checked="" type="checkbox"/> OPCIONAL	
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
ESTABILIDAD:	Durante toda la vida del sistema
DESCRIPCIÓN:	Listar los elementos de los menús alfabéticamente.

Tabla 40: Requisito UI-R007

UI-R008	
PRIORIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	FUENTE: <input type="checkbox"/> CLIENTE <input checked="" type="checkbox"/> DESARROLLADOR
NECESIDAD: <input type="checkbox"/> ESENCIAL <input checked="" type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
CLARIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
ESTABILIDAD:	Durante toda la vida del sistema
DESCRIPCIÓN:	Información y ayuda para el usuario accesible desde la propia aplicación.

Tabla 41: Requisito UI-R008

4. Diseño

4.1. Definición de la arquitectura del sistema

Para el diseño de la aplicación haremos uso de las especificaciones detalladas en el apartado 3. *Análisis* de este documento. La finalidad es definir la arquitectura del sistema y del entorno tecnológico que le va a dar soporte, junto con una especificación detallada de los componentes de información que forman parte de la misma.

La **arquitectura del sistema** seguirá las líneas de diseño establecidas por *WordPress* para la construcción de plugins para la plataforma y que hace las veces de Modelo Vista Controlador (MVC):

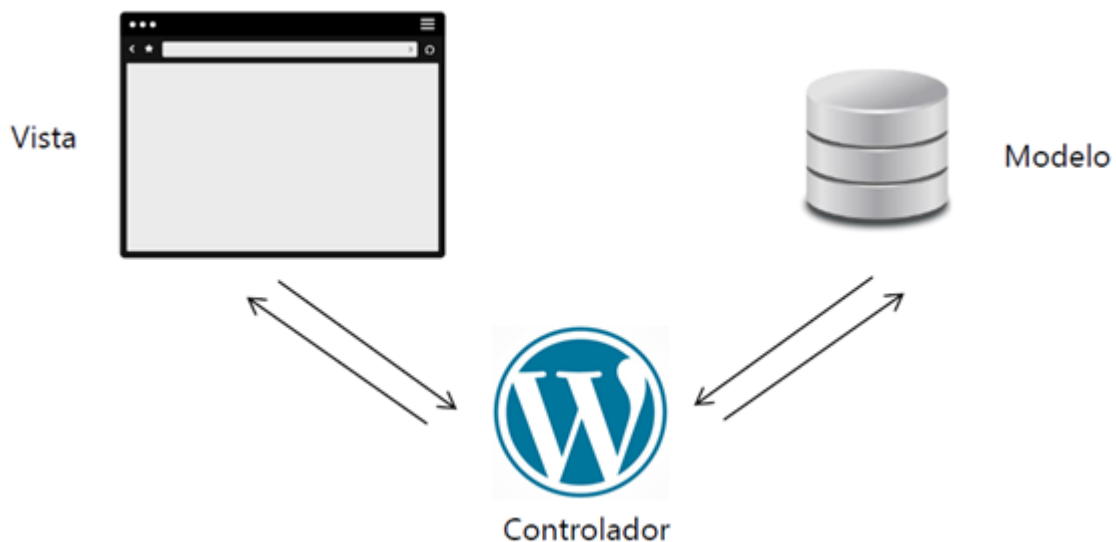


Figura 10: Modelo Vista Controlador

Así pues, el Modelo Vista Controlador divide al sistema en los siguientes subsistemas:

- **Modelo:** almacena los datos útiles del sistema. En él encontramos los datos que representan los objetos relevantes del dominio.
- **Vista:** es la parte del sistema con la que va a interactuar el usuario. Realizará la gestión de la interfaz y se comunicará con el modelo y el controlador, además de presentar la aplicación final al usuario.
- **Controlador:** la parte del sistema que recibe peticiones de la vista, recoge (o no) los datos del modelo y devuelve una respuesta al cliente. Desempeña las siguientes funciones:

- **La gestión de las reservas:** será el encargado de poner a disposición del usuario todas las reservas disponibles almacenadas en el sistema.
- **El acceso a la información:** encargado de almacenar toda la información relativa a la aplicación y de su posible modificación.
- **La gestión de actividades:** encargado de todo el proceso de modificación de las reservas.
- **Notificación:** será el controlador quién se comunice con el servidor de correo para el envío de notificaciones.

Para el diseño también es importante definir los **estándares técnicos** y de nomenclatura a seguir, así como una serie de normas y recomendaciones que se deben emplear en la construcción del sistema. Se definen pues, los siguientes elementos:

- **Idioma:** el idioma predeterminado de la aplicación será el castellano.
- **Nombre de las clases:** las clases a implementar siempre deberán empezar con mayúscula. En caso de estar formado por más de una palabra, la primera letra de cada palabra deberá empezar como mayúscula. Se podrán usar también guiones bajos como único elemento separador.
- **Nombre de las variables:** deben estar formados por letras minúsculas, números y guiones bajos. Los nombres de las variables deben ser identificativos.
- **Nombre de las constantes:** deben estar escritas en mayúsculas.
- **Métodos o funciones:** deben ser nombrados de tal manera que cada letra de las palabras que lo compongan deben ser mayúsculas y deben ir acompañados de los paréntesis.
- **Comentarios:** para escribir un comentario este debe empezar por `/*` seguido del comentario y `*/` para finalizar. También se puede escribir con `/**`. Sea como sea, deben ser legibles y aclarativos.
- **Nombre de los paquetes:** las carpetas harán las veces de paquetes, y deben indicar claramente y de manera concisa lo que contienen.

El desarrollo se realizará haciendo uso del software NetBeans IDE 7.8, una herramienta de programación de código abierto especialmente diseñada para el desarrollo de aplicaciones web. Se hará uso también de su extensión especialmente diseñada para la creación de plugins para WordPress.

Especificación de las excepciones

Es importante también tener previstos las posibles situaciones anómalas que se puedan producir en el funcionamiento y ejecución de la aplicación. En las siguientes tablas definimos los siguientes apartados:

- Tipo y descripción de la excepción.
- Estado en el que se encuentra el sistema antes de producirse la excepción.
- Elemento afectado.
- Respuesta del sistema.

EX-001	
TIPO:	Caída del Servicio de la Base de datos.
DESCRIPCIÓN:	Producida cuando se cae el servidor de la base de datos y no se pueden atender las peticiones que requieren de dicho servicio.
ESTADO:	Caída del servidor de la base de datos.
ELEMENTOS AFECTADOS:	Controlador
RESPUESTA:	El sistema mostrará un mensaje de error por pantalla, indicando al usuario que se ha perdido la conexión a Internet.

Tabla 42: Excepción EX-001

EX-002	
TIPO:	Caída del servidor Web
DESCRIPCIÓN:	Producida cuando se cae el servidor web y no se puede atender ninguna petición.
ESTADO:	Caída del servidor web
ELEMENTOS AFECTADOS:	Todos los módulos.
RESPUESTA:	El sistema dejará de funcionar de forma completa manteniendo seguros los datos.

Tabla 43: Excepción EX-002

EX-003	
TIPO:	Saturación del Sistema
DESCRIPCIÓN:	Se produce cuando el servidor está sobrecargado y no puede atender todas las peticiones en espera.
ESTADO:	Elevada cantidad de peticiones.
ELEMENTOS AFECTADOS:	Todos los módulos
RESPUESTA:	El sistema mandará un aviso de incidencia a los administradores y mostrará un mensaje de error para informar al usuario que el sistema no puede atender más peticiones

Tabla 44: Excepción EX-003

EX-004	
TIPO:	Caída del servidor de autenticación
DESCRIPCIÓN:	Producida cuando se cae el servidor LDAP de la universidad.
ESTADO:	Caída del servidor LDAP
ELEMENTOS AFECTADOS:	Controlador
RESPUESTA:	El sistema será incapaz de mostrar el nombre de los profesores que imparten las clases ni será posible que estos realicen nuevas reservas.

Tabla 45: Excepción EX-004

4.2. Diseño de la arquitectura de soporte

En cuanto a la arquitectura del sistema, además de la arquitectura de MVC se pretende diseñar varios subsistemas específicos con el objetivo de que estos sean reutilizables en la medida de lo posible. A continuación se definen los campos que se utilizarán para la descripción de los módulos de los subsistemas:

- **Identificador:** identifica de manera unívoca cada módulo. Se seguirá la siguiente nomenclatura: SSX-YY (Nombre):
 - X: podrá tomar los valores M (Modelo), V (Vista) o C (Controlador).
 - YY: conformará la numeración para la identificación del subsistema, incrementando en una unidad consecutivamente con cada nuevo subsistema, empezando por 01.
 - Nombre: denominación del módulo.
- **Propósito:** breve descripción del objetivo del módulo.
- **Funcionalidad:** descripción de las funciones que desempeña el módulo en el sistema.
- **Dependencias:** módulos adicionales necesarios para el funcionamiento.
- **Recursos:** sistemas externos necesarios para el correcto funcionamiento del módulo.
- **Subordinados:** otros módulos que emplean la funcionalidad proporcionada por este.

SSV – 01 (Módulo de calendario)	
PROPÓSITO	Se encarga de mostrar las reservas disponibles en forma de vista de calendario semanal y diario.
FUNCIONALIDAD	Mostrar reservas de hoy Mostrar reservas por día Mostrar reservas por aula
DEPENDENCIAS	Módulo de acceso a la información
RECURSOS	Ninguno
SUBORDINADOS	Ninguno

Tabla 46: Subsistema SSV-01 (Módulo de calendario)

Análisis, diseño e implantación de una aplicación para la gestión de reservas para las aulas docentes del departamento de informática

SSM – 02 (Módulo de acceso a la información)	
PROPÓSITO	Encargado de gestionar la información sobre reservas y configuración del sistema.
FUNCIONALIDAD	Servir la información sobre las reservas Servir la información sobre las aulas Servir la información sobre las asignaturas Servir la información sobre los grados Servir la información sobre el software Servir la información sobre las máquinas virtuales Servir la información de las opciones.
DEPENDENCIAS	Ninguno
RECURSOS	Base de datos MySQL
SUBORDINADOS	Ninguno

Tabla 47: Subsistema SSM-02 (Módulo de acceso a la información)

SSC – 03 (Módulo de usuario)	
PROPÓSITO	Se encarga de la gestión de inicio de sesión y realización de reservas por parte de los profesores.
FUNCIONALIDAD	Inicio de sesión para profesores Conexión con el servidor LDAP Realizar reservas por los profesores
DEPENDENCIAS	Ninguno
RECURSOS	Servidor LDAP de la Universidad
SUBORDINADOS	Ninguno

Tabla 48: Subsistema SSC-03 (Módulo de usuario)

SSC – 04 (Módulo de administración)	
PROPÓSITO	Encargado de la gestión de todo el panel de administración.
FUNCIONALIDAD	Mostrar el menú de administración Ver, añadir y borrar reservas Ver, añadir y borrar aulas Ver, añadir y borrar asignaturas Ver, añadir y borrar grados Ver, añadir y borrar software Ver, añadir y borrar máquinas virtuales Ver y descargar informes de ocupación Ver y editar las opciones de configuración
DEPENDENCIAS	Módulo de acceso a la información
RECURSOS	Ninguno
SUBORDINADOS	Módulo de gestión de reservas Módulo de creación de reservas Módulo de gestión de aulas Módulo de gestión de asignaturas Módulo de gestión de grados Módulo de gestión de software Módulo de gestión de máquinas virtuales Módulo de gestión de informes Módulo de gestión de opciones

Tabla 49: Subsistema SSC-04 (Módulo de administración)

SSV – 05 (Módulo de gestión de reservas)	
PROPÓSITO	Se encarga de la gestión de las reservas
FUNCIONALIDAD	Ver las reservas Editar las reservas Borrar las reservas Confirmar las reservas pendientes
DEPENDENCIAS	Módulo de acceso a la información
RECURSOS	Ninguno
SUBORDINADOS	Ninguno

Tabla 50: Subsistema SSV-05 (Módulo de gestión de reservas)

SSV – 06 (Módulo de creación de reservas)	
PROPÓSITO	Encargado de la gestión del formulario de creación de reservas
FUNCIONALIDAD	Crear reservas
DEPENDENCIAS	Ninguno
RECURSOS	Ninguno
SUBORDINADOS	Ninguno

Tabla 51: Subsistema SSV-06 (Módulo de creación de reservas)

SSV – 07 (Módulo de gestión de aulas)	
PROPÓSITO	Se encarga de la gestión de las aulas
FUNCIONALIDAD	Añadir aulas Ver las aulas Editar las aulas Borrar las aulas
DEPENDENCIAS	Módulo de acceso a la información
RECURSOS	Ninguno
SUBORDINADOS	Ninguno

Tabla 52: Subsistema SSV-07 (Módulo de gestión de aulas)

SSV – 08 (Módulo de gestión de asignaturas)	
PROPÓSITO	Encargado de la gestión de las asignaturas
FUNCIONALIDAD	Añadir asignaturas Ver las asignaturas Editar las asignaturas Borrar las asignaturas
DEPENDENCIAS	Módulo de acceso a la información
RECURSOS	Ninguno
SUBORDINADOS	Ninguno

Tabla 53: Subsistema SSV-08 (Módulo de gestión de asignaturas)

SSV – 09 (Módulo de gestión de grados)	
PROPÓSITO	Se encarga de la gestión de los grados
FUNCIONALIDAD	Añadir grados Ver los grados Editar los grados Borrar los grados
DEPENDENCIAS	Módulo de acceso a la información
RECURSOS	Ninguno
SUBORDINADOS	Ninguno

Tabla 54: Subsistema SSV-09 (Módulo de gestión de grados)

SSV – 10 (Módulo de gestión de software)	
PROPÓSITO	Encargado de la gestión del software
FUNCIONALIDAD	Añadir software Ver el software Editar el software Borrar el software
DEPENDENCIAS	Módulo de acceso a la información
RECURSOS	Ninguno
SUBORDINADOS	Ninguno

Tabla 55: Subsistema SSV-10 (Módulo de gestión de software)

SSV – 11 (Módulo de gestión de máquinas virtuales)	
PROPÓSITO	Se encarga de la gestión de las máquinas virtuales
FUNCIONALIDAD	Añadir máquinas virtuales Ver las máquinas virtuales Editar las máquinas virtuales Borrar las máquinas virtuales
DEPENDENCIAS	Módulo de acceso a la información
RECURSOS	Ninguno
SUBORDINADOS	Ninguno

Tabla 56: Subsistema SSV-11 (Módulo de gestión de máquinas virtuales)

SSV – 12 (Módulo de gestión de informes)	
PROPÓSITO	Encargado de la gestión de los informes
FUNCIONALIDAD	Generar informes de ocupación de las aulas Descargar informes de ocupación de las aulas
DEPENDENCIAS	Módulo de acceso a la información
RECURSOS	Ninguno
SUBORDINADOS	Ninguno

Tabla 57: Subsistema SSV-12 (Módulo de gestión de informes)

SSV – 13 (Módulo de gestión de opciones)	
PROPÓSITO	Se encarga de la gestión de las opciones
FUNCIONALIDAD	Ver las configuración seleccionada Editar la configuración Mostrar ayuda
DEPENDENCIAS	Módulo de acceso a la información
RECURSOS	Ninguno
SUBORDINADOS	Ninguno

Tabla 58: Subsistema SSV-13 (Módulo de gestión de opciones)

4.3. Diseño de casos de uso

Con los módulos del sistema especificados podemos comenzar con el diseño de los **casos de uso** desde el punto de vista técnico. Con el fin de facilitar su comprensión haremos uso de diagramas de secuencia para cada uno de los casos:

CU_01	
CASO DE USO	Ver las reservas
ACTOR	Usuario
OBJETIVO	El usuario selecciona un aula o día de la semana y comprueba las reservas asignadas.
PRECONDICIONES	Se haya seleccionado un aula o día
POSTCONDICIONES	Se muestran las reservas

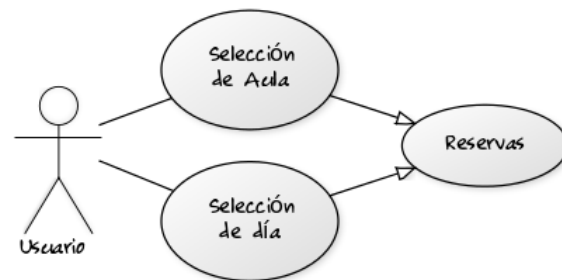


Figura 11: Caso de Uso CU_01

Tabla 59: Caso de Uso CU_01

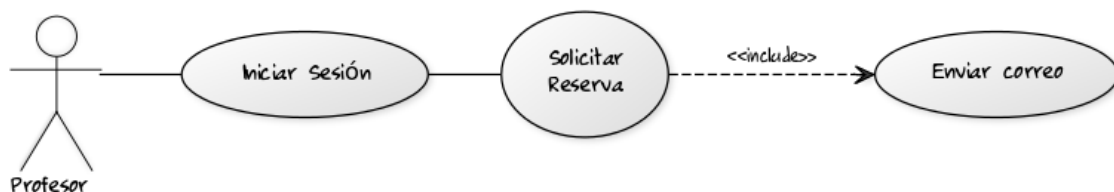


Figura 12: Caso de Uso CU_02

CU_02	
CASO DE USO	Realizar una solicitud de reserva
ACTOR	Usuario (profesor)
OBJETIVO	El profesor completa el formulario de solicitud de reserva
PRECONDICIONES	Haber iniciado sesión
POSTCONDICIONES	Se enviará un correo al mail del laboratorio con la información de la reserva solicitada

Tabla 60: Caso de Uso CU_02

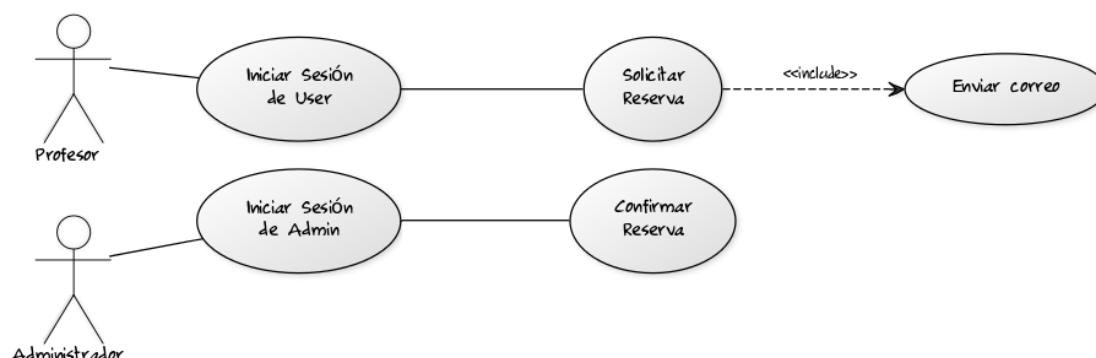


Figura 13: Caso de Uso CU_03

CU_03	
CASO DE USO	Confirmar una reserva
ACTOR	Administrador
OBJETIVO	El administrador confirma una reserva solicitada por un profesor
PRECONDICIONES	Haber iniciado sesión como administrador, que se haya solicitado una reserva
POSTCONDICIONES	Se añade la reserva solicitada a la base de datos

Tabla 61: Caso de Uso CU_03



Figura 14: Caso de Uso CU_04

CU_04	
CASO DE USO	Añadir una reserva
ACTOR	Administrador
OBJETIVO	El administrador añade una nueva reserva al sistema
PRECONDICIONES	Haber iniciado sesión
POSTCONDICIONES	Se añade la reserva a la base de datos

Tabla 62: Caso de Uso CU_04



Figura 15: Caso de Uso CU_05


CU_05	
CASO DE USO	Gestionar reservas
ACTOR	Administrador
OBJETIVO	El administrador realiza la gestión de las reservas
PRECONDICIONES	Haber iniciado sesión como administrador
POSTCONDICIONES	Se aplican las modificaciones solicitadas por el administrador


Tabla 63: Caso de Uso CU_06

4.4. Funcionamiento y taxonomía de la interfaz

La **interfaz del sistema** diseñada debe permitir a los usuarios del sistema realizar todas las funciones definidas anteriormente en los casos de uso de manera sencilla y cómoda. Analicemos en profundidad la interfaz propuesta:

Pantalla de ocupación diaria


**LABORATORIO DEL
DEPARTAMENTO DE INFORMÁTICA**


**Universidad
Carlos III de Madrid**

[INICIO](#)
[INFORMACIÓN](#)
[SERVICIOS](#)
[NOTICIAS](#)
[MAPA WEB](#)
[BUSCADOR](#)
[CONTACTAR](#)

Laboratorio del Departamento de Informática // Información // Ocupación de las aulas // Ocupación diaria

A A A

Ocupación diaria

Estas son las reservas programadas para el 23/03/2015

Hora	Aula 1	Aula 2	Aula 3	Aula 4	Aula 5
09:00					
09:15					
09:30					
09:45	Asignatura 1 (G01) Grado 1				Asignatura 3 (G11) Grado 3
10:00					
10:15					
10:30					
10:45					
11:00					
11:15					
11:30					
11:45					
12:00					
12:15					
12:30					
12:45					
13:00					
13:15					
13:30					
13:45					
14:00				Asignatura 4 (G16) Grado 4	
14:15					
14:30					
14:45					
15:00					
15:15					
15:30					
15:45					
16:00					
16:15					
16:30					
16:45					
17:00					
17:15					
17:30					
17:45					
18:00					
18:15					
18:30					
18:45					
19:00					
19:15					
19:30					
19:45					
20:00					
20:15					
20:30					
20:45					

RESERVAS PUNTUALES

Asignatura	Profesor	Aula	Horario
Asignatura 4 (G16)	JAIME PONS BAILLY-BAILLIERE	Aula 3	De 13:00 a 15:00

Ocupación

- Ocupación diaria
- Aula 2.2.C05
- Aula 2.2.C06
- Aula 4.0.F16
- Aula 4.0.F18

Esta semana

- Lunes
- Martes
- Miércoles
- Jueves
- Viernes

Semana próxima

- Lunes de la semana que viene
- Martes de la semana que viene
- Miércoles de la semana que viene
- Jueves de la semana que viene
- Viernes de la semana que viene

Menú Principal

- Ocupación de las aulas
- Horario de las Aulas
- Apertura de cuentas
- Tutoriales
- F.A.Q.

Última Hora

Sample Post 6 13 mayo, 2013
Sample Post 5 13 mayo, 2013
Sample Post 4 13 mayo, 2013


Figura 16: Pantalla de ocupación diaria


La página de ocupación diaria mostrará información sobre las reservas programadas para un día determinado y consta de los siguientes elementos:

PANTALLA DE OCUPACIÓN DIARIA			
ID	NOMBRE	TIPO	MÉTODOS IMPLICADOS ONCLICK
A	Reservas	Texto	-
B	Reservas Puntuales	Tabla	-
C	Barra de Menús	Texto	1. Selección del día del que ver las reservas. 2. Navegación a la pantalla de ocupación por aulas. 3. Navegación a otras secciones de la web.

Tabla 64: Pantalla de ocupación diaria

Pantalla de ocupación por aulas


**LABORATORIO DEL
DEPARTAMENTO DE INFORMÁTICA**


**Universidad
Carlos III de Madrid**

[INICIO](#)
[INFORMACIÓN](#)
[SERVICIOS](#)
[NOTICIAS](#)
[MAPA WEB](#)
[BUSCADOR](#)
[CONTACTAR](#)

Laboratorio del Departamento de Informática // Información // Ocupación de las aulas // Aula 2.2.C05

A A A

Aula 2.2.C05

Hora	Lunes	Martes	Miércoles	Jueves	Viernes
09:00					
09:15					
09:30					
09:45					
10:00		Reserva puntual			Reserva puntual
10:15					
10:30					
10:45					
11:00					
11:15					
11:30					
11:45					
12:00	Reserva puntual				
12:15					
12:30					
12:45					
13:00					
13:15					
13:30					
13:45					
14:00					
14:15					
14:30					
14:45					
15:00					
15:15					
15:30					
15:45					
16:00					
16:15					
16:30					
16:45					
17:00					
17:15					
17:30					
17:45					
18:00					
18:15					
18:30					
18:45					
19:00					
19:15					
19:30					
19:45					
20:00					
20:15					
20:30					
20:45					

RESERVAS PUNTUALES

Asignatura	Profesor	Fecha	Horario
Asignatura 5 (G21)	PRUEBA PRUEBA	01/09/2015	De 09:00 a 11:00
Asignatura 2 (G06)	PRUEBA PRUEBA	01/09/2015	De 09:00 a 11:00
Asignatura 1 (G01)	PRUEBA PRUEBA	01/09/2015	De 11:00 a 13:00

Ocupación

- Ocupación diaria
- Aula 2.2.C05
- Aula 2.2.C06
- Aula 4.0.F16
- Aula 4.0.F18

Esta semana

- Lunes
- Martes
- Miércoles
- Jueves
- Viernes

Semana próxima

- Lunes de la semana que viene
- Martes de la semana que viene
- Miércoles de la semana que viene
- Jueves de la semana que viene
- Viernes de la semana que viene

Menú Principal

- Ocupación de las aulas
- Horario de las Aulas
- Apertura de cuentas
- Tutoriales
- F.A.Q.

Última Hora

Sample Post 6 13 mayo, 2013
Sample Post 5 13 mayo, 2013
Sample Post 4 13 mayo, 2013

Copyright © 2015. Laboratorio del Departamento de Informática.

Departamento de Informática Universidad Carlos III

Figura 17: Pantalla de ocupación por aulas

Análisis, diseño e implantación de una aplicación para la gestión de reservas para las aulas docentes del departamento de informática

La página de ocupación por aulas mostrará información sobre las reservas programadas durante toda la semana para un aula determinada y consta de los siguientes elementos:

PANTALLA DE OCUPACIÓN POR AULA			
ID	NOMBRE	TIPO	MÉTODOS IMPLICADOS ONCLICK
A	Reservas	Texto	-
B	Reservas Puntuales	Tabla	-
C	Barra de Menús	Texto	1. Selección del día del que ver las reservas. 2. Navegación a la pantalla de ocupación por aulas. 3. Navegación a otras secciones de la web.

Tabla 65: Pantalla de ocupación por aulas

Pantalla de login

Reservas

Bienvenido al sistema de reserva de aulas del Laboratorio del Departamento de Informática de La Universidad Carlos III de Madrid. Desde esta aplicación usted podrá realizar su solicitud para de reserva de alguna de las aulas que gestiona el laboratorio.

Para entrar, inicie sesión con su correo de pas.

Usuario

Contraseña



Figura 18: Pantalla de login

La pantalla de login permite a los usuarios acceder a la solicitud de reservas de las aulas y consiste en un sencillo y habitual formulario de inicio de sesión:

PANTALLA DE LOGIN			
ID	NOMBRE	TIPO	MÉTODOS IMPLICADOS ONCLICK
A	Inicio de Sesión	Formulario	1. Iniciar Sesión

Tabla 66: Pantalla de login

Pantalla de solicitud de reservas

Reservas

Solicitar reserva

Bienvenido al sistema de solicitud de reservas para las aulas del Laboratorio del Departamento de Informática. El siguiente formulario le ayudará a tramitar su reserva de la forma más sencilla posible:

Profesor

Nombre del profesor que impartirá la asignatura o, en su defecto, del coordinador de la asignatura.

Asignatura

Asignatura que se impartirá durante la sesión de la reserva.

Tipo de reserva

Indica la periodicidad de la reserva: la puntual solo se hará efectiva para un único día, mientras que la cuatrimestral reserva el aula para todo el cuatrimestre.

Fecha

Indique la fecha en que se desea realizar la reserva.

El usuario no tiene permisos para acceder a esta página

Figura 19: Pantalla de solicitud de reservas 1

Reservas

Paso 2 de 3: Selección de horario

A continuación se muestra la ocupación de las aulas los Miércoles:

Hora	Aula 1	Aula 2	Aula 3	Aula 4	Aula 5
09:00					
09:15					
09:30					
09:45					
10:00					
10:15					
10:30					
10:45					
11:00					
11:15					
11:30					
11:45					
12:00					
12:15					
12:30					
12:45					
13:00					
13:15					
13:30					
13:45					
14:00					
14:15					
14:30					
14:45					
15:00					
15:15					

Asignatura 1 (G03) Grado 1

Asignatura 2 (G06) Grado 2

Menú Principal

- Ocupación de las aulas
- Horario de las Aulas
- Apertura de cuentas
- Tutoriales
- F.A.Q.

Última Hora

Sample Post 6 13 mayo, 2013
Sample Post 5 13 mayo, 2013
Sample Post 4 13 mayo, 2013

Grupo

Grupo de la asignatura que va a aprovechar dicha reserva.

Aula

Aula que se desea reservar. Tenga en cuenta el número de puestos de trabajo disponibles en cada aula a la hora de realizar la reserva.

Hora
 -

Hora en la que se desea realizar la reserva

Figura 20: Pantalla de solicitud de reservas 2



Reservas

Paso 3 de 3: Confirmar datos

Profesor: 100292713
 Asignatura: Asignatura 3
 Grupo: 11
 Tipo de Reserva: cuatrimestral
 Día: Todos los Miércoles del 1 cuatrimestre
 Aula: Aula 1
 Horario: de 09:00 a 11:00

Para comprobar que todos los datos introducidos son correctos y realizar la petición de su reserva haga click en "Enviar solicitud". Tenga en cuenta que haber solicitado la reserva no significa que esta le haya sido concedida. El equipo del Laboratorio del Departamento de Informática se pondrá en contacto con usted en breve para comunicarle la resolución de su reserva.

[Atrás](#) [Enviar Solicitud](#)

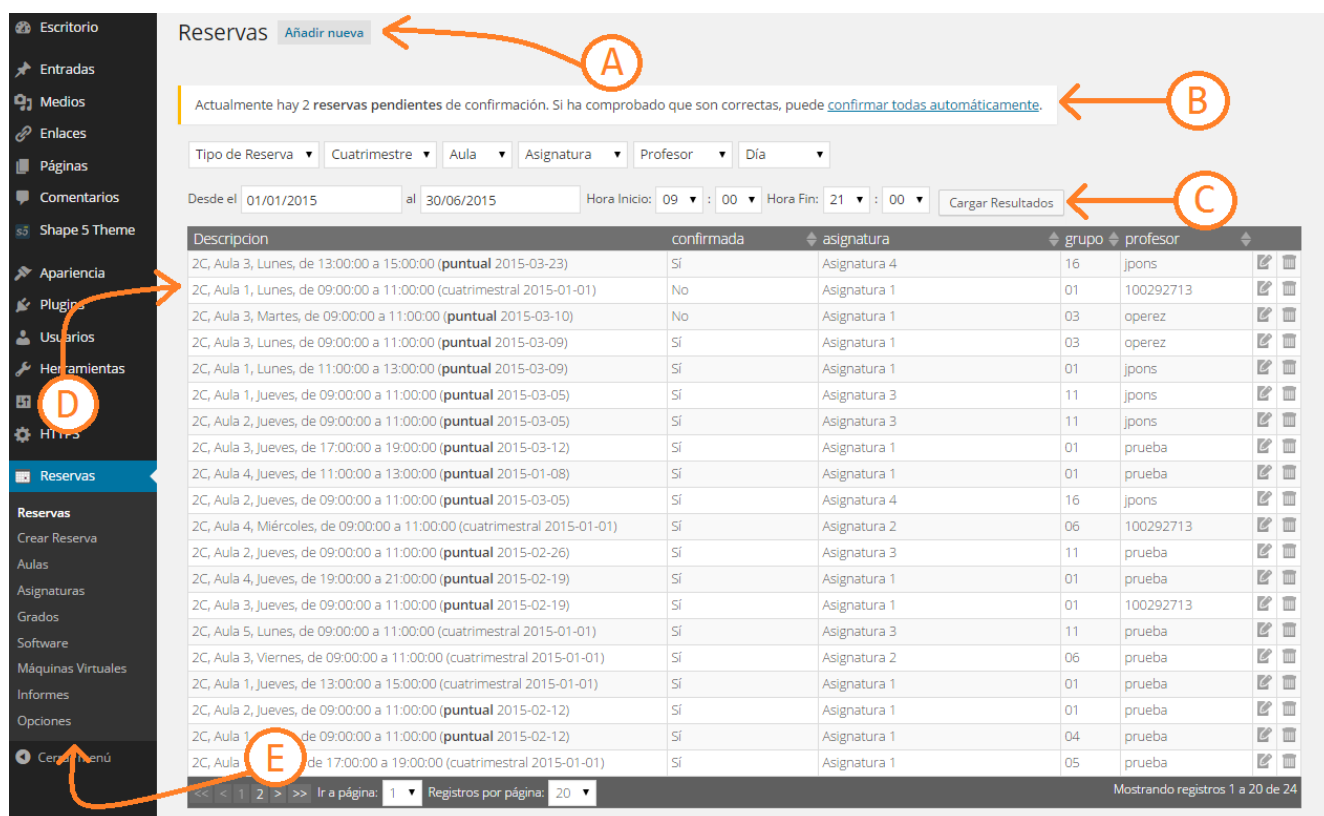
Figura 21: Pantalla de solicitud de reservas 3

La pantalla de solicitud de reservas consiste en un formulario de tres páginas donde ha de rellenarse la información relativa a la reserva:

PANTALLA DE SOLICITUD DE RESERVA			
ID	NOMBRE	TIPO	MÉTODOS IMPLICADOS ONCLICK
A	Reservas	Texto	-
B	Menú de reserva	Tabla	-
C	Resumen de reserva	Texto	-
D	Enviar Solicitud	Botón	1. Añadir reserva a la base de datos. 2. Enviar correo al administrador del sistema.

Tabla 67: Pantalla de solicitud de reservas

Pantalla de gestión de reservas



Reservas [Añadir nueva](#)

Actualmente hay 2 reservas pendientes de confirmación. Si ha comprobado que son correctas, puede [confirmar todas automáticamente](#).

Tipo de Reserva: Cuatrimestre Aula Asignatura Profesor Día

Desde el: 01/01/2015 al: 30/06/2015 Hora Inicio: 09 : 00 Hora Fin: 21 : 00 Cargar Resultados

Descripcion	confirmada	asignatura	grupo	profesor
2C, Aula 3, Lunes, de 13:00:00 a 15:00:00 (puntual 2015-03-23)	Sí	Asignatura 4	16	jpons
2C, Aula 1, Lunes, de 09:00:00 a 11:00:00 (cuatrimestral 2015-01-01)	No	Asignatura 1	01	100292713
2C, Aula 3, Martes, de 09:00:00 a 11:00:00 (puntual 2015-03-10)	No	Asignatura 1	03	operez
2C, Aula 3, Lunes, de 09:00:00 a 11:00:00 (puntual 2015-03-09)	Sí	Asignatura 1	03	operez
2C, Aula 1, Lunes, de 11:00:00 a 13:00:00 (puntual 2015-03-09)	Sí	Asignatura 1	01	jpons
2C, Aula 1, Jueves, de 09:00:00 a 11:00:00 (puntual 2015-03-05)	Sí	Asignatura 3	11	jpons
2C, Aula 2, Jueves, de 09:00:00 a 11:00:00 (puntual 2015-03-05)	Sí	Asignatura 3	11	jpons
2C, Aula 3, Jueves, de 17:00:00 a 19:00:00 (puntual 2015-03-12)	Sí	Asignatura 1	01	prueba
2C, Aula 4, Jueves, de 11:00:00 a 13:00:00 (puntual 2015-01-08)	Sí	Asignatura 1	01	prueba
2C, Aula 2, Jueves, de 09:00:00 a 11:00:00 (puntual 2015-03-05)	Sí	Asignatura 4	16	jpons
2C, Aula 4, Miércoles, de 09:00:00 a 11:00:00 (cuatrimestral 2015-01-01)	Sí	Asignatura 2	06	100292713
2C, Aula 2, Jueves, de 09:00:00 a 11:00:00 (puntual 2015-02-26)	Sí	Asignatura 3	11	prueba
2C, Aula 4, Jueves, de 19:00:00 a 21:00:00 (puntual 2015-02-19)	Sí	Asignatura 1	01	prueba
2C, Aula 3, Jueves, de 09:00:00 a 11:00:00 (puntual 2015-02-19)	Sí	Asignatura 1	01	100292713
2C, Aula 5, Lunes, de 09:00:00 a 11:00:00 (cuatrimestral 2015-01-01)	Sí	Asignatura 3	11	prueba
2C, Aula 3, Viernes, de 09:00:00 a 11:00:00 (cuatrimestral 2015-01-01)	Sí	Asignatura 2	06	prueba
2C, Aula 1, Jueves, de 13:00:00 a 15:00:00 (cuatrimestral 2015-01-01)	Sí	Asignatura 1	01	prueba
2C, Aula 2, Jueves, de 09:00:00 a 11:00:00 (puntual 2015-02-12)	Sí	Asignatura 1	01	prueba
2C, Aula 1, de 09:00:00 a 11:00:00 (puntual 2015-02-12)	Sí	Asignatura 1	04	prueba
2C, Aula, de 17:00:00 a 19:00:00 (cuatrimestral 2015-01-01)	Sí	Asignatura 1	05	prueba

Mostrando registros 1 a 20 de 24

Figura 22: Pantalla de gestión de reservas

La pantalla de gestión de reservas permite a los administradores editar o eliminar las reservas existentes para mantener una gestión y planificación eficientes. Consta de los siguientes elementos:

PANTALLA DE GESTIÓN DE RESERVAS			
ID	NOMBRE	TIPO	MÉTODOS IMPLICADOS ONCLICK
A	Crear Reserva	Botón	1. Ir a la pantalla de creación de reservas
B	Alerta de pendientes	Texto	1. Confirmar todas las reservas pendientes
C	Filtrar	Formulario	1. Filtrar la lista de reservas según los parámetros
D	Tabla de Reservas	Tabla	1. Editar una reserva 2. Eliminar una reserva
E	Menú de Reservas	Menú	1. Navegar a las distintas opciones de la aplicación

Tabla 68: Pantalla de gestión de reservas

Pantalla de creación de reservas

Crear Reserva

Crear una nueva reserva: Paso 1 de 3

Profesor

Correo electrónico del profesor que realiza la reserva.

Asignatura

Asignatura 1 ▼

Asignatura que se impartirá durante la sesión de la reserva.

Tipo de reserva

Puntual ▼

Indica la periodicidad de la reserva: la puntual solo se hará efectiva para un único día, mientras que la cuatrimestral reserva el aula para todo el cuatrimestre.

Fecha

Indique la fecha en que se desea realizar la reserva.

Siguiente

Figura 23: Pantalla de creación de reservas 1

Escritorio

Entradas

Medios

Enlaces

Páginas

Comentarios

Shape 5 Theme

Apariencia

Plugins

Usuarios

Herramientas

Ajustes

HTTPS

Reservas

Reservas

Crear Reserva

Aulas

Asignaturas

Grados

Software

Máquinas Virtuales

Informes

Opciones

Cerrar menú

Crear Reserva

Selección de aula y horario: Paso 2 de 3

A continuación se muestra la ocupación de las aulas el día 2015-03-23:

Hora	Aula 1	Aula 2	Aula 3	Aula 4	Aula 5
09:00					
09:15					
09:30					
09:45	Asignatura 1 (G01) Grado 1	Asignatura 5 (G21) Grado 5 el 2015-09-01			Asignatura 3 (G11) Grado 3
10:00					
10:15					
10:30					
10:45					
11:00					
11:15					
11:30					
11:45					
12:00					
12:15					
12:30					
12:45					
13:00					
13:15					
13:30					
13:45					
14:00			Asignatura 4 (G16) Grado 4 el 2015-03-23		
14:15					
14:30					
14:45					
15:00					

A la izquierda puede verse la ocupación de las aulas del día en el que se desea programar la reserva. Por favor, compruebe que el horario elegido se corresponde con alguno de los huecos libres entre las reservas ya establecidas y complete los siguientes campos:

Grupo

11 ▼

Grupo de la asignatura que va a aprovechar dicha reserva.

Aula

Aula 1 ▼

Aula que se desea reservar. Tenga en cuenta el número de puestos de trabajo disponibles en cada aula.

Hora

09:00 - 11:00

Hora en la que se desea realizar la reserva.

Atrás Siguiente

Figura 24: Pantalla de creación de reservas 2

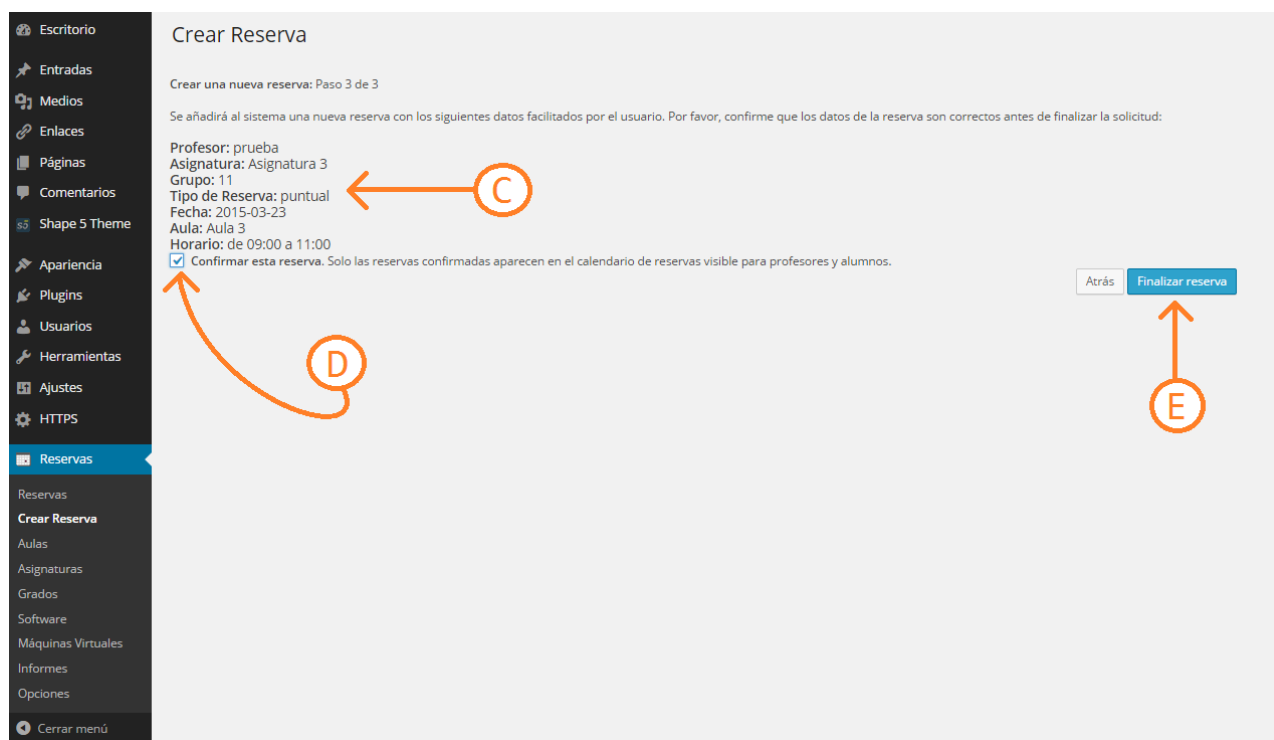


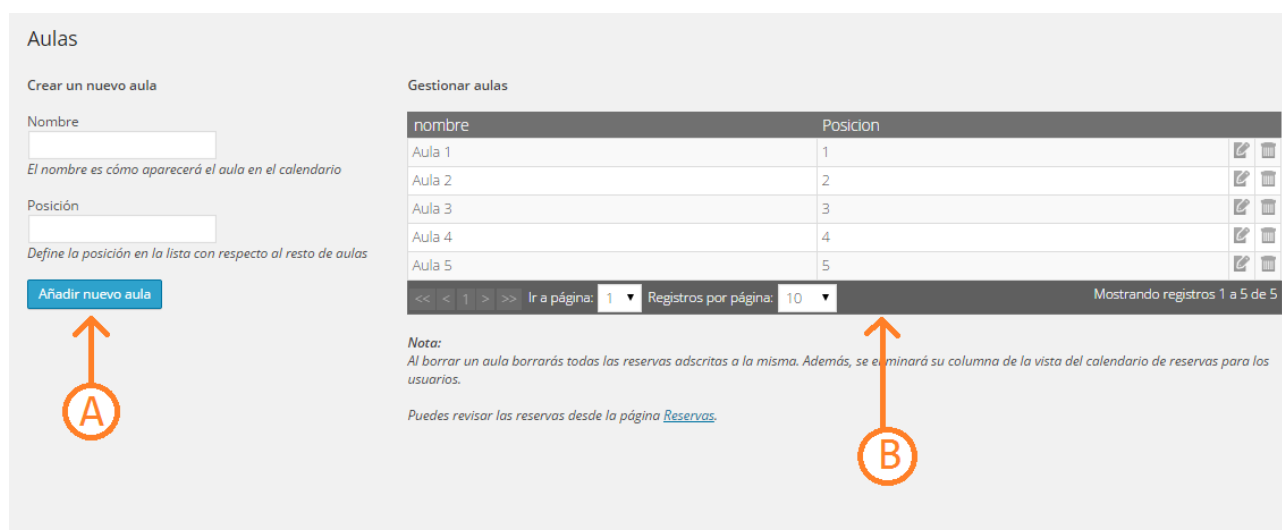
Figura 25: Pantalla de creación de reservas 3

Al igual que la pantalla de solicitud de reserva, la creación de reservas está dividida en tres pasos diferentes que permiten añadir una nueva reserva a la base de datos de la aplicación. Consta de las siguientes partes:

PANTALLA DE CREACIÓN DE RESERVAS			
ID	NOMBRE	TIPO	MÉTODOS IMPLICADOS ONCLICK
A	Reservas	Tabla	-
B	Botón siguiente	Botón	1. Lleva al paso final
C	Resumen	Texto	1. Resumen de los datos de la reserva
D	Confirmar reserva	Checkbox	1. Marcar la nueva reserva como confirmada
E	Finalizar Reserva	Botón	1. Añade la reserva a la base de datos

Tabla 69: Pantalla de creación de reservas

Pantalla de gestión de aulas



Aulas

Crear un nuevo aula

Nombre

El nombre es cómo aparecerá el aula en el calendario

Posición

Define la posición en la lista con respecto al resto de aulas

Añadir nuevo aula

Gestionar aulas

nombre	Posición		
Aula 1	1		
Aula 2	2		
Aula 3	3		
Aula 4	4		
Aula 5	5		

<< < 1 > >> Ir a página: 1 Registros por página: 10 Mostrando registros 1 a 5 de 5

Nota:
Al borrar un aula borrarás todas las reservas adscritas a la misma. Además, se eliminará su columna de la vista del calendario de reservas para los usuarios.

Puedes revisar las reservas desde la página [Reservas](#).

Figura 26: Pantalla de gestión de aulas

La pantalla de gestión de aulas permite añadir nuevas aulas al sistema de gestión de reservas además de editar o eliminar las aulas ya existentes. Consta de los siguientes elementos:

PANTALLA DE GESTIÓN DE AULAS			
ID	NOMBRE	TIPO	MÉTODOS IMPLICADOS ONCLICK
A	Añadir Aula	Formulario	1. Añadir un nuevo aula
B	Tabla de aulas	Tabla	1. Editar o eliminar aulas

Tabla 70: Pantalla de gestión de aulas

Pantalla de gestión de asignaturas

Asignatura

Crear una nueva asignatura

Nombre

El nombre es cómo aparecerá la asignatura en la BBDD

Grado

Corresponde con la titulación a la que pertenece la asignatura

Curso

El año de la titulación en que se cursa la asignatura

Cuatrimestre

Cuatrimestre para el que se imparte la asignatura según el plan de la titulación.

Software

☐ Software 1

☐ Software 2

☐ Software 3

☐ Software 4

☐ Software 5

El software utilizado durante las clases de dicha asignatura

Máquinas Virtuales

☐ Máquina Virtual 1

☐ Máquina Virtual 2

☐ Máquina Virtual 3

☐ Máquina Virtual 4

☐ Máquina Virtual 5

Las máquinas virtuales utilizadas durante las clases

Gestionar asignaturas

Nombre	Grado	Cuatrimestre	Curso	
Asignatura 5	Grado 5	2	3	

Software requerido

Nombre	Version	
Software 1	1	
Software 2	2	
Software 3	3	
Software 4	4	
Software 5	5	

Asignatura 4	Grado 4	2	4	
Asignatura 3	Grado 3	1	3	
Asignatura 2	Grado 2	2	2	
Asignatura 1	Grado 1	1	1	

<< < 1 > >> Ir a página: Registros por página: Mostrando registros 1 a 5 de 5

Nota:
Al borrar una asignatura borrarás de forma permanente todas las reservas asociadas a la misma. Sin embargo, no se verán afectados ni el software ni las máquinas virtuales usadas para impartirla.

Puedes revisar las Reservas disponibles en la página [Reservas](#).

A

B

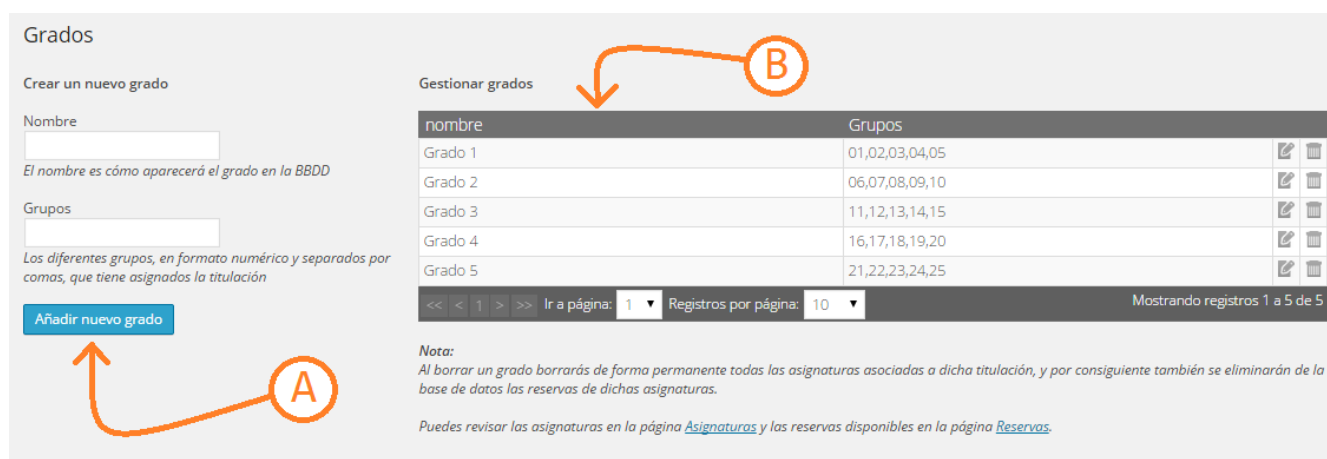
Figura 27: Pantalla de gestión de asignaturas

La pantalla de gestión de asignaturas permite añadir nuevas asignaturas al sistema de gestión de reservas además de editar o eliminar las asignaturas ya existentes. Consta de los siguientes elementos:

PANTALLA DE GESTIÓN DE ASIGNATURAS			
ID	NOMBRE	TIPO	MÉTODOS IMPLICADOS ONCLICK
A	Añadir Asignatura	Formulario	1. Añadir una nueva asignatura
B	Tabla de asignaturas	Tabla	1. Editar o eliminar asignaturas

Tabla 71: Pantalla de gestión de asignaturas

Pantalla de gestión de grados



Grados

Crear un nuevo grado

Nombre

El nombre es cómo aparecerá el grado en la BBDD

Grupos

Los diferentes grupos, en formato numérico y separados por comas, que tiene asignados la titulación

Añadir nuevo grado

Gestionar grados

nombre	Grupos		
Grado 1	01,02,03,04,05		
Grado 2	06,07,08,09,10		
Grado 3	11,12,13,14,15		
Grado 4	16,17,18,19,20		
Grado 5	21,22,23,24,25		

<< < 1 > >> Ir a página: Registros por página: Mostrando registros 1 a 5 de 5

Nota:
 Al borrar un grado borrarás de forma permanente todas las asignaturas asociadas a dicha titulación, y por consiguiente también se eliminarán de la base de datos las reservas de dichas asignaturas.

Puedes revisar las asignaturas en la página [Asignaturas](#) y las reservas disponibles en la página [Reservas](#).

Figura 28: Pantalla de gestión de grados

La pantalla de gestión de grados permite añadir nuevos grados al sistema de gestión de reservas además de editar o eliminar los grados ya existentes. Consta de los siguientes elementos:

PANTALLA DE GESTIÓN DE GRADOS			
ID	NOMBRE	TIPO	MÉTODOS IMPLICADOS ONCLICK
A	Añadir Grado	Formulario	1. Añadir un nuevo grado
B	Tabla de grados	Tabla	1. Editar o eliminar grados

Tabla 72: Pantalla de gestión de grados

Pantalla de gestión de software

Software

Crear un nuevo software

Nombre

El nombre para identificar el programa

Versión

La versión del software instalada en las aulas

Url

Dirección a la web oficial de descarga del software

Descripción

Información sobre qué hace el programa

Fecha de instalación

El día en el que se instaló el software

Instrucciones de instalación

Instrucciones para la correcta instalación

[Añadir nuevo software](#)

Gestionar software

nombre	version	url	descripcion	Fecha Instalación	Instalación
Software 1	1	www.software1.es	Descripción 1	2015-02-12	Instrucciones 1
Software 2	2	www.software2.es	Descripción 2	2015-02-12	Instrucciones 2
Software 3	3	www.software3.es	Descripción 3	2015-02-12	Instrucciones 3
Software 4	4	www.software4.es	Descripción 4	2015-02-12	Instrucciones 4
Software 5	5	www.software5.com	Descripción 5	2015-02-12	Instalación 5

<< < 1 > >> Ir a página: Registros por página: Mostrando registros 1 a 5 de 5

Nota:
Al borrar un software no borrarás aquellas asignaturas que lo usan. En su lugar, se eliminarán las referencias a dicho software en todas las asignaturas que alguna vez lo utilizaron.

Puedes revisar las asignaturas y el software que utilizan en la página [Asignaturas](#).

Figura 29: Pantalla de gestión de software

La pantalla de gestión de software permite añadir nuevos grados al sistema de gestión de reservas además de editar o eliminar el software ya existente. Consta de los siguientes elementos:

PANTALLA DE GESTIÓN DE SOFTWARE			
ID	NOMBRE	TIPO	MÉTODOS IMPLICADOS ONCLICK
A	Añadir Software	Formulario	1. Añadir un nuevo software
B	Tabla de software	Tabla	1. Editar o eliminar software

Tabla 73: Pantalla de gestión de software

Pantalla de gestión de máquinas virtuales

Máquinas Virtuales

Crear una nueva máquina virtual

Nombre

El nombre para identificar el programa

Sistema Operativo

El sistema operativo instalado en la máquina virtual

Aplicaciones

Aplicaciones instaladas en la máquina virtual

Capacidad











Espacio en disco que ocupa la máquina virtual

Aulas

Aulas en las que está instalada la máquina

Añadir nueva máquina virtual

Gestionar máquinas virtuales

Nombre	Sistema Operativo	Aplicaciones	Capacidad	Aulas	
Máquina Virtual 1	Sistema Operativo 1	Aplicación 1	1GB	Aula 1	 
Máquina Virtual 2	Sistema Operativo 2	Aplicación 2	2GB	Aula 2	 
Máquina Virtual 3	Sistema Operativo 3	Aplicación 3	3GB	Aula 3	 
Máquina Virtual 4	Sistema Operativo 4	Aplicación 4	4GB	Aula 4	 
Máquina Virtual 5	Sistema Operativo 5	Aplicación 5	5GB	Aula 5	 

<< < 1 > >> Ir a página: Registros por página: Mostrando registros 1 a 5 de 5

Nota:
 Al borrar una máquina virtual no borrarás aquellas asignaturas que la usan. En su lugar, se eliminarán las referencias a dicha máquina virtual en todas las asignaturas que alguna vez la utilizaron.

Puedes revisar las asignaturas y las máquinas virtuales que utilizan en la página [Asignaturas](#).

Figura 30: Pantalla de gestión de máquinas virtuales

La pantalla de gestión de máquinas virtuales permite añadir nuevas máquinas virtuales al sistema de gestión de reservas además de editar o eliminar las máquinas virtuales ya existentes. Consta de los siguientes elementos:

PANTALLA DE GESTIÓN DE MÁQUINAS VIRTUALES			
ID	NOMBRE	TIPO	MÉTODOS IMPLICADOS ONCLICK
A	Añadir MV	Formulario	1. Añadir una nueva máquina virtual
B	Tabla de MV	Tabla	1. Editar o eliminar máquinas virtuales

Tabla 74: Pantalla de gestión de máquinas virtuales

Pantalla de gestión de informes

Informes

March 23, 2015, 11:12 am

Puede volver a ver las estadísticas de cualquier curso del que haya registros en la base de datos
2015

Primer Cuatrimestre

Asignatura	Aula	Horas
Asignatura 1		14
Asignatura 1	Aula 4	14
Asignatura 2		14
Asignatura 5		14
Asignatura 5	Aula 2	28
	Total	84

Segundo Cuatrimestre

Asignatura	Aula	Horas
Asignatura 1	Aula 1	102
Asignatura 1	Aula 2	2
Asignatura 1	Aula 3	8
Asignatura 1	Aula 4	4
Asignatura 2	Aula 3	14
Asignatura 2	Aula 4	14
Asignatura 3	Aula 1	2
Asignatura 3	Aula 2	4
Asignatura 3	Aula 5	14
Asignatura 4	Aula 2	2
Asignatura 4	Aula 3	2
	Total	168

Descargar este informe

Puede descargar este informe como un documento de hoja de cálculo (.xls) de Microsoft Office Excel. Tenga en cuenta que los informes realizados para el presente curso son provisionales y podrían cambiar si se solicitan nuevas reservas puntuales o cuatrimestrales por parte del equipo docente de la Universidad.

Descargar

Figura 31: Pantalla de gestión de informes

La pantalla de gestión de informes proporciona información interesante sobre la ocupación de las aulas para cada año con datos almacenados. Adicionalmente también permite descargar los informes mostrados en pantalla en un documento Excel.

PANTALLA DE GESTIÓN DE INFORMES			
ID	NOMBRE	TIPO	MÉTODOS IMPLICADOS ONCLICK
A	Selección de año	Desplegable	1. Mostrar los informes del año seleccionado
B	Tabla de informes	Tabla	-
C	Descargar informe	Botón	1. Descargar el informe en .xls

Tabla 75: Pantalla de gestión de informes

Pantalla de opciones

Opciones

Cuatrimestres

Los cuatrimestres indican qué clases están activas en cada momento del curso. Así, las clases "cuatrimestrales" que estén reservadas para el primer cuatrimestre solo serán visibles para los usuarios durante el periodo estimado para dicho cuatrimestre. Además, durante el periodo de verano solo se mostrarán las reservas puntuales.

Primer cuatrimestre	Desde el	<input type="text" value="01/09"/>	al	<input type="text" value="31/12"/>
Segundo cuatrimestre	Desde el	<input type="text" value="01/01"/>	al	<input type="text" value="30/06"/>
Verano	Desde el	<input type="text" value="01/07"/>	al	<input type="text" value="31/08"/>

Estilo


El calendario de reservas utiliza un sistema de colores para distinguir entre las reservas puntuales y las cuatrimestrales. Estos colores pueden ser modificados para adaptarlos al tema de Wordpress que usted esté utilizando para su web.

Color para reservas cuatrimestrales	<input type="color" value="#0000FF"/>
Color para reservas puntuales	<input type="color" value="#8B4513"/>

Conexión Ldap

La autenticación para el acceso al panel de usuario se realiza siempre a través de un servidor LDAP. Para que esto se realice de forma correcta, es necesario que configure correctamente los parámetros de su conexión con el servidor.

Usuario	<input type="text"/>
Contraseña	<input type="password"/>
Servidor Ldap	<input type="text" value="ldaps://ldap.uc3m.es"/>
Base DN	<input type="text" value="ou=Personal,ou=Gente,o=I"/>

← 

Ayuda

Shortcodes

Para el correcto uso del plugin es necesaria la utilización de códigos abreviados o shortcodes de Wordpress. Estos códigos sirven para invocar funciones que ejecuten distintas funcionalidades del plugin y muestren el contenido del mismo a los usuarios. Los distintos shortcodes que puede utilizar con este plugin son:

Reservas de hoy	[reservas]
Reservas del día	[reservas dia="(Lunes),(Martes,etc)"]
Reservas por aula	[reservas aula="nombre aula"]
Panel de usuario	[login]

Gracias por crear con [WordPress](#).

Versión 4.1.1

Figura 32: Pantalla de opciones

La pantalla de opciones contiene el formulario de configuración de los distintos parámetros del gestor de reservas así como una pequeña sección con ayuda e información sobre el uso de la aplicación:

PANTALLA DE GESTIÓN DE OPCIONES			
ID	NOMBRE	TIPO	MÉTODOS IMPLICADOS ONCLICK
A	Guardar Cambios	Formulario	1. Actualiza los parámetros de opciones en la base de datos

Tabla 76: Pantalla de gestión de informes

4.5. Diseño de clases

El buen **diseño de clases** determinará en gran medida la estructura de nuestra aplicación y su sencillez a la hora de realizar cambios en el futuro. Este es un factor muy importante debido a que tras la implantación el proceso de mantenimiento y actualización del sistema correrá a cargo de personal diferente al que desarrolló la aplicación, siendo vital un correcto diseño de clases. Cada uno de los subsistemas que conforman el gestor de reservas contiene las siguientes clases:

Clases de lógica de calendario	
CALENDAR	Implementa la funcionalidad de visualización de reservas en un calendario

Tabla 77: Clases de lógica de calendario

Clases de acceso a la información	
DATABASE	Implementa funcionalidades relacionadas con la persistencia de datos

Tabla 78: Clases de acceso a la información

Clases de lógica de usuario	
USER	Implementa la funcionalidad de inicio de sesión y solicitud de reservas

Tabla 79: Clases de lógica de usuario

Clases de lógica de administración	
ADMINISTRATOR	Implementa la funcionalidad del menú de administración

Tabla 80: Clases de lógica de administración

Clases de lógica de gestión de reservas	
GEST_SCHEDULE	Implementa la funcionalidad de ver, confirmar, editar y borrar reservas.
TABLE_SCHEDULE	Implementa la construcción de la tabla de visualización de reservas

Tabla 81: Clases de lógica de gestión de reservas

Clases de lógica de creación de reservas	
CREATE_SCHEDULE	Implementa la funcionalidad de creación de reservas

Tabla 82: Clases de lógica de creación de reservas

Clases de lógica de gestión de aulas	
GEST_CLASSROOM	Implementa la funcionalidad de ver, añadir, editar y borrar aulas.
TABLE_CLASSROOM	Implementa la construcción de la tabla de visualización de aulas

Tabla 83: Clases de lógica de gestión de aulas

Clases de lógica de gestión de asignaturas	
GEST_COURSE	Implementa la funcionalidad de ver, añadir, editar y borrar asignaturas.
TABLE_COURSE	Implementa la construcción de la tabla de visualización de asignaturas
TABLE_MV_COURSE	Implementa la construcción de la tabla de máquinas virtuales de asignaturas
TABLE_SW_COURSE	Implementa la construcción de la tabla de software de asignaturas

Tabla 84: Clases de lógica de gestión de asignaturas

Clases de lógica de gestión de grados	
GEST_DEGREE	Implementa la funcionalidad de ver, añadir, editar y borrar grados
TABLE_DEGREE	Implementa la construcción de la tabla de visualización de grados

Tabla 85: Clases de lógica de gestión de grados

Clases de lógica de gestión de software	
GEST_SOFTWARE	Implementa la funcionalidad de ver, añadir, editar y borrar software
TABLE_SOFTWARE	Implementa la construcción de la tabla de visualización de software

Tabla 86: Clases de lógica de gestión de software

Clases de lógica de gestión de máquinas virtuales	
GEST_DEGREE	Implementa la funcionalidad de ver, añadir, editar y borrar máquinas virtuales
TABLE_DEGREE	Implementa la construcción de la tabla de visualización de máquinas virtuales

Tabla 87: Clases de lógica de gestión de máquinas virtuales

Clases de lógica de gestión de informes	
GEST_DATA	Implementa la funcionalidad de ver los informes de ocupación
GENERATE_EXCEL	Implementa la descarga de los informes en formato XLS

Tabla 88: Clases de lógica de gestión de informes

Clases de lógica de gestión de opciones	
GEST_OPTIONS	Implementa la funcionalidad de ver y editar las opciones de configuración

Tabla 89: Clases de lógica de gestión de opciones

Todas estas clases están relacionadas de forma jerárquica en la cual cada una realiza una función específica y bien diferenciada. El siguiente diagrama identifica las clases de diseño que forman la aplicación junto con sus relaciones:

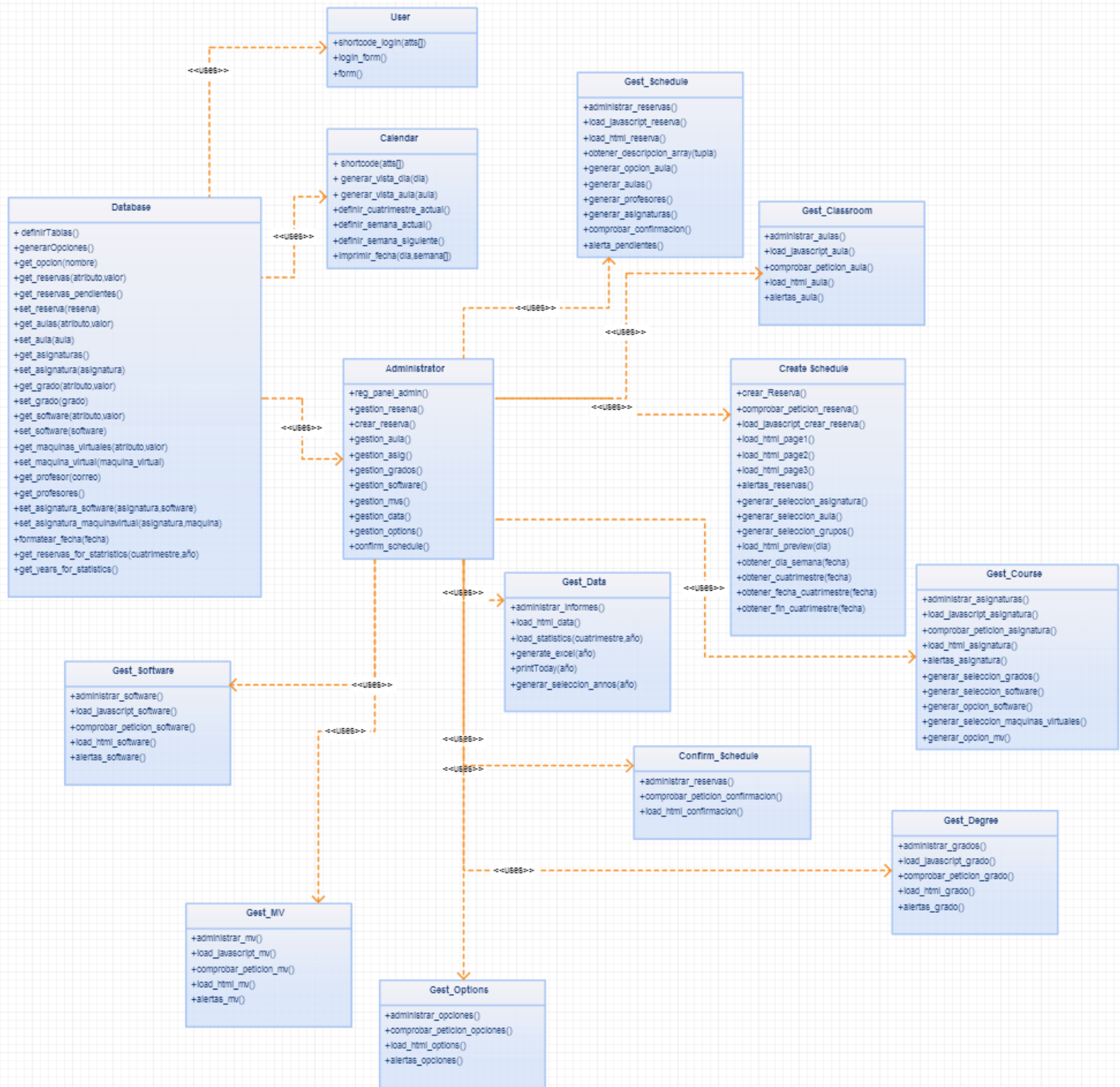


Figura 33: Diagrama de clases

Además, cada clase contendrá a su vez uno o varios métodos que son necesarios para el desempeño de la función que realizan:

Clase Calendar	
<i>shortcode (atts[])</i>	
DESCRIPCIÓN	Comprueba si alguna página quiere mostrar información sobre las reservas
ARGUMENTOS	Array con las opciones que se desean mostrar
FUNCIONAMIENTO	Leerá las opciones contenidas en <i>atts[]</i> y llamará a los métodos encargados de componer la vista de calendario en función de los mismos
RETORNO	Código HTML a mostrar
<i>generar_vista_dia (String dia)</i>	
DESCRIPCIÓN	Genera la vista de reservas diarias
ARGUMENTOS	Nombre del día de la semana
FUNCIONAMIENTO	Calculará la fecha en función del día de la semana y obtendrá las reservas programadas de la base de datos para construir el código HTML de la vista de calendario
RETORNO	Código HTML de la vista de calendario
<i>generar_vista_aula (String aula)</i>	
DESCRIPCIÓN	Genera la vista de reservas de la semana para un aula determinada
ARGUMENTOS	Nombre del aula
FUNCIONAMIENTO	Calculará las fechas de los días de la semana actual y obtendrá las reservas programadas de la base de datos para construir el código HTML de la vista de calendario
RETORNO	Código HTML de la vista de calendario
<i>definir_cuatrimestre_actual ()</i>	
DESCRIPCIÓN	Determina el cuatrimestre en el que nos encontramos
ARGUMENTOS	-
FUNCIONAMIENTO	Comparará la fecha actual con la de inicio y fin de los cuatrimestres para determinar el cuatrimestre en este momento
RETORNO	Cuatrimestre actual
<i>definir_semana_actual ()</i>	
DESCRIPCIÓN	Determina la fecha en formato YYYY-mm-dd de los días de la semana
ARGUMENTOS	-
FUNCIONAMIENTO	Comprueba la fecha actual y el día de la semana y genera el resto de fechas para cada día a partir de ellos
RETORNO	Fechas de toda la semana
<i>definir_semana_siguiente ()</i>	
DESCRIPCIÓN	Determina la fecha en formato YYYY-mm-dd de los días de la semana próxima
ARGUMENTOS	-
FUNCIONAMIENTO	Comprueba la fecha actual y el día de la semana y genera las fechas para cada día de la semana próxima a partir de ellos
RETORNO	Fechas de toda la semana próxima

<i>imprimir_fecha (String dia, semana[])</i>	
DESCRIPCIÓN	Obtiene la fecha de las reservas que se quieren mostrar y la formatea al estilo dd-mm-YYYY
ARGUMENTOS	Día de la semana, fechas de la semana
FUNCIONAMIENTO	Comprueba el día de la semana, obtiene la fecha y la formatea al estilo dd-mm-YYY
RETORNO	String con la fecha solicitada

Tabla 90: Clase Calendar

Clase Database	
<i>definirTablas ()</i>	
DESCRIPCIÓN	Crea las tablas en la base de datos
ARGUMENTOS	-
FUNCIONAMIENTO	Construye la sentencia SQL y la ejecuta
RETORNO	-
<i>generarOpciones ()</i>	
DESCRIPCIÓN	Genera los valores por defecto de las opciones del sistema
ARGUMENTOS	-
FUNCIONAMIENTO	Inserta los valores por defecto en la tabla de la base de datos
RETORNO	-
<i>get_opcion (String nombre)</i>	
DESCRIPCIÓN	Obtiene el valor de una opción determinada
ARGUMENTOS	Nombre de la opción que se desea recuperar
FUNCIONAMIENTO	Se consulta en la base de datos el valor de la opción con ese nombre
RETORNO	Valor de la opción solicitada
<i>get_reservas(String atributo, String valor)</i>	
DESCRIPCIÓN	Recupera las reservas cuyo atributo sea igual a un valor dado
ARGUMENTOS	Nombre del valor que se desea comparar y dicho valor
FUNCIONAMIENTO	Recupera de la base de datos todas las reservas si cumplen que el atributo que se ha pasado como parámetro tenga el valor deseado.
RETORNO	Lista de reservas
<i>get_reservas_pendientes()</i>	
DESCRIPCIÓN	Recupera todas las reservas pendientes
ARGUMENTOS	-
FUNCIONAMIENTO	Recupera todas las reservas pendientes de la base de datos.
RETORNO	Lista de reservas pendientes
<i>set_reserva(confirmada, asignatura, grupo, profesor, aula, tipo_reserva, dia, cuatrimestre, fecha, hora_inicio, hora_fin)</i>	
DESCRIPCIÓN	Almacena una nueva reserva en la base de datos
ARGUMENTOS	Los parámetros de la reserva a almacenar
FUNCIONAMIENTO	Recoge todos los valores e inserta una nueva reserva en la base de datos con los parámetros deseados
RETORNO	El id de la reserva creada

<i>get_aulas(String atributo, String valor)</i>	
DESCRIPCIÓN	Recupera las aulas cuyo atributo sea igual a un valor dado
ARGUMENTOS	Nombre del valor que se desea comparar y dicho valor
FUNCIONAMIENTO	Recupera de la base de datos todas las aulas si cumplen que el atributo que se ha pasado como parámetro tenga el valor deseado
RETORNO	Lista de aulas
<i>set_aula(nombre, posicion)</i>	
DESCRIPCIÓN	Almacena un nuevo aula en la base de datos
ARGUMENTOS	Los parámetros del aula a almacenar
FUNCIONAMIENTO	Recoge todos los valores e inserta un nuevo aula en la base de datos con los parámetros deseados
RETORNO	-
<i>get_asignaturas()</i>	
DESCRIPCIÓN	Recupera todas las asignaturas almacenadas en el sistema
ARGUMENTOS	-
FUNCIONAMIENTO	Recupera de la base de datos todas las asignaturas
RETORNO	Lista de asignaturas
<i>set_asignatura(nombre,grado,curso,cuatrimestre,software,maquinas)</i>	
DESCRIPCIÓN	Almacena una nueva asignatura en la base de datos
ARGUMENTOS	Los parámetros de la asignatura a almacenar
FUNCIONAMIENTO	Recoge todos los valores e inserta una nueva asignatura en la base de datos con los parámetros deseados
RETORNO	-
<i>get_grado(String atributo, String valor)</i>	
DESCRIPCIÓN	Recupera los grados cuyo atributo sea igual a un valor dado
ARGUMENTOS	Nombre del valor que se desea comparar y dicho valor
FUNCIONAMIENTO	Recupera de la base de datos todos los grados si cumplen que el atributo que se ha pasado como parámetro tenga el valor deseado
RETORNO	Lista de grados
<i>set_grado(nombre,grupos)</i>	
DESCRIPCIÓN	Almacena un nuevo grado en la base de datos
ARGUMENTOS	Los parámetros del grado a almacenar
FUNCIONAMIENTO	Recoge todos los valores e inserta un nuevo grado en la base de datos con los parámetros deseados
RETORNO	-
<i>get_software(String atributo, String valor)</i>	
DESCRIPCIÓN	Recupera el software cuyo atributo sea igual a un valor dado
ARGUMENTOS	Nombre del valor que se desea comparar y dicho valor
FUNCIONAMIENTO	Recupera de la base de datos todo el software si cumple que el atributo que se ha pasado como parámetro tenga el valor deseado
RETORNO	Lista de software
<i>set_software(nombre,version,url,descripcion,fecha,instalacion)</i>	
DESCRIPCIÓN	Almacena un nuevo software en la base de datos
ARGUMENTOS	Los parámetros del software a almacenar
FUNCIONAMIENTO	Recoge todos los valores e inserta un nuevo software en la base de datos con los parámetros deseados
RETORNO	-

<i>get_maquinas_virtuales(String atributo, String valor)</i>	
DESCRIPCIÓN	Recupera las máquinas virtuales cuyo atributo sea igual a un valor dado
ARGUMENTOS	Nombre del valor que se desea comparar y dicho valor
FUNCIONAMIENTO	Recupera de la base de datos todas las máquinas virtuales si cumplen que el atributo que se ha pasado como parámetro tenga el valor deseado
RETORNO	Lista de máquinas virtuales
<i>set_maquina_virtual(nombre,so,aplicaciones,capacidad,aula)</i>	
DESCRIPCIÓN	Almacena una nueva máquina virtual en la base de datos
ARGUMENTOS	Los parámetros de la máquina virtual a almacenar
FUNCIONAMIENTO	Recoge todos los valores e inserta una nueva máquina virtual en la base de datos con los parámetros deseados
RETORNO	-
<i>get_profesor(String correo)</i>	
DESCRIPCIÓN	Recupera el nombre de un profesor a partir de su correo electrónico
ARGUMENTOS	Correo electrónico del profesor
FUNCIONAMIENTO	Conecta con el LDAP para obtener el nombre del profesor en función del correo electrónico pasado como parámetro.
RETORNO	Nombre completo del profesor
<i>get_profesores()</i>	
DESCRIPCIÓN	Obtiene el correo electrónico de todos los profesores que dan clase en las aulas
ARGUMENTOS	-
FUNCIONAMIENTO	Recupera de la base de datos todos los profesores que han dado alguna clase en las aulas
RETORNO	Lista de correos electrónicos de profesores
<i>set_asignatura_software(asignatura,software)</i>	
DESCRIPCIÓN	Almacena en la base de datos la relación entre una asignatura y el software que usa
ARGUMENTOS	Id de asignatura e Id del software que utiliza
FUNCIONAMIENTO	Almacena en la tabla intermedia una nueva entrada con los datos proporcionados
RETORNO	-
<i>set_asignatura_maquinavirtual(asignatura,maquina)</i>	
DESCRIPCIÓN	Almacena en la base de datos la relación entre una asignatura y la máquina virtual que usa
ARGUMENTOS	Id de asignatura e Id de la máquina virtual que utiliza
FUNCIONAMIENTO	Almacena en la tabla intermedia una nueva entrada con los datos proporcionados
RETORNO	-
<i>formatear_fecha(valor)</i>	
DESCRIPCIÓN	Convierte el valor del día de la semana del formato inglés (Monday) al formato castellano (Lunes)
ARGUMENTOS	Día de la semana en inglés
FUNCIONAMIENTO	Compara el valor pasado como parámetro y lo traduce
RETORNO	Día de la semana en castellano
<i>get_reservas_for_statistics(int cuatrimestre, int anno)</i>	
DESCRIPCIÓN	Recupera las reservas de la base de datos de un cuatrimestre y un año determinados
ARGUMENTOS	Cuatrimestre y año que se desea recuperar
FUNCIONAMIENTO	Recupera las reservas de la base de datos de un cuatrimestre y un año determinados para usarlos en el cálculo de las estadísticas
RETORNO	Lista de reservas

<i>get_years_for_statistics()</i>	
DESCRIPCIÓN	Recupera los años en los que se realizaron reservas
ARGUMENTOS	-
FUNCIONAMIENTO	Comprueba cuantos años se han realizado reservas en la base de datos viendo la fecha de las reservas almacenadas
RETORNO	Lista de años

Tabla 91: Clase Database

Clase User	
<i>shortcode_login(atts[])</i>	
DESCRIPCIÓN	Comprueba si alguna página quiere mostrar el panel de usuario
ARGUMENTOS	Array con las opciones que se desean mostrar
FUNCIONAMIENTO	Leerá las opciones contenidas en <i>atts[]</i> y llamará a los métodos encargados de componer la vista del panel de usuario en función de los mismos
RETORNO	Código HTML a mostrar
<i>login_form()</i>	
DESCRIPCIÓN	Genera el formulario de inicio de sesión
ARGUMENTOS	-
FUNCIONAMIENTO	Compone el código HTML necesario para mostrar el panel de inicio de sesión
RETORNO	Código HTML del formulario de inicio de sesión
<i>form()</i>	
DESCRIPCIÓN	Genera el formulario de solicitud de reservas
ARGUMENTOS	-
FUNCIONAMIENTO	Compone el código HTML necesario para el formulario de solicitud de reservas
RETORNO	Código HTML del formulario de solicitud de reservas

Tabla 92: Clase User

Clase Administrator	
<i>reg_panel_admin()</i>	
DESCRIPCIÓN	Se encarga de registrar el panel de administración en Wordpress
ARGUMENTOS	-
FUNCIONAMIENTO	Registra la estructura del menú mediante llamadas del sistema de Wordpress para que este se integre en el menú de administración del CMS
RETORNO	-
<i>gestión_reserva()</i>	
DESCRIPCIÓN	Llama al módulo de gestión de reservas
ARGUMENTOS	-
FUNCIONAMIENTO	Permite que se cree el enlace en el menú de administración a la página de gestión de reservas
RETORNO	-
<i>crear_reserva()</i>	
DESCRIPCIÓN	Llama al módulo de creación de reservas
ARGUMENTOS	-
FUNCIONAMIENTO	Permite que se cree el enlace en el menú de administración a la página de creación de reservas
RETORNO	-

<i>gestión_aula()</i>	
DESCRIPCIÓN	Llama al módulo de gestión de aulas
ARGUMENTOS	-
FUNCIONAMIENTO	Permite que se cree el enlace en el menú de administración a la página de gestión de aulas
RETORNO	-
<i>gestión_asig()</i>	
DESCRIPCIÓN	Llama al módulo de gestión de asignaturas
ARGUMENTOS	-
FUNCIONAMIENTO	Permite que se cree el enlace en el menú de administración a la página de gestión de asignaturas
RETORNO	-
<i>gestión_grados()</i>	
DESCRIPCIÓN	Llama al módulo de gestión de grados
ARGUMENTOS	-
FUNCIONAMIENTO	Permite que se cree el enlace en el menú de administración a la página de gestión de grados
RETORNO	-
<i>gestión_software()</i>	
DESCRIPCIÓN	Llama al módulo de gestión de software
ARGUMENTOS	-
FUNCIONAMIENTO	Permite que se cree el enlace en el menú de administración a la página de gestión de software
RETORNO	-
<i>gestión_mvs()</i>	
DESCRIPCIÓN	Llama al módulo de gestión de máquinas virtuales
ARGUMENTOS	-
FUNCIONAMIENTO	Permite que se cree el enlace en el menú de administración a la página de gestión de máquinas virtuales
RETORNO	-
<i>gestión_data()</i>	
DESCRIPCIÓN	Llama al módulo de gestión de informes
ARGUMENTOS	-
FUNCIONAMIENTO	Permite que se cree el enlace en el menú de administración a la página de gestión de informes
RETORNO	-
<i>gestión_options()</i>	
DESCRIPCIÓN	Llama al módulo de gestión de opciones
ARGUMENTOS	-
FUNCIONAMIENTO	Permite que se cree el enlace en el menú de administración a la página de gestión de opciones
RETORNO	-

<i>confirm_schedule()</i>	
DESCRIPCIÓN	Llama al módulo de confirmación de reservas
ARGUMENTOS	-
FUNCIONAMIENTO	Crea la página de confirmación de reservas llamando al módulo correspondiente
RETORNO	-

Tabla 93: Clase Administrator

Clase Gest_Schedule	
<i>administrar_reservas()</i>	
DESCRIPCIÓN	Conforma toda la vista de la página de gestión de reservas
ARGUMENTOS	-
FUNCIONAMIENTO	Hace diversas llamadas al resto de métodos de la clase para conformar la vista de la página
RETORNO	-
<i>load_javascript_reserva()</i>	
DESCRIPCIÓN	Genera el código javascript necesario para componer la vista
ARGUMENTOS	-
FUNCIONAMIENTO	Hace las llamadas a funciones necesarias y compone el código javascript de la página
RETORNO	-
<i>load_html_reserva()</i>	
DESCRIPCIÓN	Genera el código HTML necesario para componer la vista
ARGUMENTOS	-
FUNCIONAMIENTO	Hace las llamadas a funciones necesarias y compone el código HTML de la página
RETORNO	-
<i>obtener_descripcion_array(tupla)</i>	
DESCRIPCIÓN	Genera la descripción de la reserva en función de sus atributos
ARGUMENTOS	Tupla de la reserva almacenada en la base de datos
FUNCIONAMIENTO	A partir de los atributos pasados se genera la descripción de la reserva
RETORNO	Descripción de la reserva
<i>generar_opcion_aula()</i>	
DESCRIPCIÓN	Obtiene la lista de aulas como un desplegable HTML
ARGUMENTOS	-
FUNCIONAMIENTO	Recupera las aulas de la base de datos y compone el código HTML del desplegable
RETORNO	Código HTML de la lista
<i>generar_aulas()</i>	
DESCRIPCIÓN	Obtiene la lista de aulas como un desplegable HTML
ARGUMENTOS	-
FUNCIONAMIENTO	Recupera las aulas de la base de datos y compone el código HTML del desplegable
RETORNO	Código HTML de la lista
<i>generar_profesores()</i>	
DESCRIPCIÓN	Obtiene la lista de profesores como un desplegable HTML
ARGUMENTOS	-
FUNCIONAMIENTO	Recupera los profesores de la base de datos y compone el código HTML del desplegable
RETORNO	Código HTML de la lista

<i>generar_asignaturas()</i>	
DESCRIPCIÓN	Obtiene la lista de asignaturas como un desplegable HTML
ARGUMENTOS	-
FUNCIONAMIENTO	Recupera las asignaturas de la base de datos y compone el código HTML del desplegable
RETORNO	Código HTML de la lista
<i>comprobar_confirmacion()</i>	
DESCRIPCIÓN	Comprueba si se ha realizado una confirmación y muestra un aviso
ARGUMENTOS	-
FUNCIONAMIENTO	Comprueba si se ha confirmado una reserva pendiente y en caso afirmativo muestra un aviso en pantalla
RETORNO	-
<i>alerta_pendientes()</i>	
DESCRIPCIÓN	Comprueba si hay reservas pendientes y muestra un mensaje de aviso
ARGUMENTOS	-
FUNCIONAMIENTO	Comprueba si actualmente hay reservas pendientes en la base de datos y en caso afirmativo muestra un mensaje de aviso
RETORNO	-

Tabla 94: Clase Gest_Schedule

Clase Create_Schedule	
<i>crearReserva()</i>	
DESCRIPCIÓN	Conforma toda la vista de la página de creación de reservas
ARGUMENTOS	-
FUNCIONAMIENTO	Hace diversas llamadas al resto de métodos de la clase para conformar la vista de la página
RETORNO	-
<i>comprobar_peticion_reserva()</i>	
DESCRIPCIÓN	Genera el código HTML del formulario de creación de reservas
ARGUMENTOS	-
FUNCIONAMIENTO	Comprueba el proceso de creación y genera las distintas páginas en consecuencia
RETORNO	-
<i>load_javascript_crear_reserva()</i>	
DESCRIPCIÓN	Genera el código javascript necesario para componer la vista
ARGUMENTOS	-
FUNCIONAMIENTO	Hace las llamadas a funciones necesarias y compone el código javascript de la página
RETORNO	-
<i>load_html_page1()</i>	
DESCRIPCIÓN	Genera la página 1 del formulario de creación de reservas
ARGUMENTOS	-
FUNCIONAMIENTO	Genera el código HTML de la página 1 del formulario para la creación de reservas
RETORNO	-

<i>load_html_page2()</i>	
DESCRIPCIÓN	Genera la página 2 del formulario de creación de reservas
ARGUMENTOS	-
FUNCIONAMIENTO	Genera el código HTML de la página 2 del formulario para la creación de reservas
RETORNO	-
<i>load_html_page3()</i>	
DESCRIPCIÓN	Genera la página 3 del formulario de creación de reservas
ARGUMENTOS	-
FUNCIONAMIENTO	Genera el código HTML de la página 3 del formulario para la creación de reservas
RETORNO	-
<i>alertas_reservas()</i>	
DESCRIPCIÓN	Comprueba si se ha creado una nueva reserva y muestra una notificación
ARGUMENTOS	-
FUNCIONAMIENTO	Comprueba si se ha creado una nueva reserva y en caso afirmativo muestra una notificación en pantalla
RETORNO	-
<i>generar_seleccion_asignatura()</i>	
DESCRIPCIÓN	Obtiene la lista de asignaturas como un desplegable HTML
ARGUMENTOS	-
FUNCIONAMIENTO	Recupera las asignaturas de la base de datos y compone el código HTML del desplegable
RETORNO	Código HTML de la lista
<i>generar_seleccion_aula()</i>	
DESCRIPCIÓN	Obtiene la lista de aulas como un desplegable HTML
ARGUMENTOS	-
FUNCIONAMIENTO	Recupera las aulas de la base de datos y compone el código HTML del desplegable
RETORNO	Código HTML de la lista
<i>generar_seleccion_grupos()</i>	
DESCRIPCIÓN	Obtiene la lista de grupos como un desplegable HTML
ARGUMENTOS	-
FUNCIONAMIENTO	Recupera los grupos de la base de datos y compone el código HTML del desplegable
RETORNO	Código HTML de la lista
<i>load_html_preview(String dia)</i>	
DESCRIPCIÓN	Muestra la vista previa de reservas programadas para un día
ARGUMENTOS	Día del que se quieren saber las reservas
FUNCIONAMIENTO	Recupera las reservas de la base de datos para ese día y las muestra en una vista de calendario
RETORNO	-

<i>obtener_dia_semana(Date fecha)</i>	
DESCRIPCIÓN	Devuelve el día de la semana en función de una fecha dada
ARGUMENTOS	Fecha en formato YYYY-mm-dd
FUNCIONAMIENTO	Obtiene el nombre del día de la semana en castellano en función de una fecha dada
RETORNO	Día de la semana
<i>obtener_cuatrimestre(Date fecha)</i>	
DESCRIPCIÓN	Obtiene el cuatrimestre al que pertenece una fecha dada
ARGUMENTOS	Fecha en formato YYYY-mm-dd
FUNCIONAMIENTO	Obtiene el cuatrimestre en función de la fecha dada y los periodos determinados en las opciones de configuración
RETORNO	Cuatrimestre
<i>obtener_fecha_cuatrimestre(Date fecha)</i>	
DESCRIPCIÓN	Obtiene la fecha de inicio del cuatrimestre a partir de una fecha
ARGUMENTOS	Fecha en formato YYYY-mm-dd
FUNCIONAMIENTO	Obtiene el cuatrimestre en función de la fecha dada y los periodos determinados en las opciones de configuración y determina el día de inicio de ese cuatrimestre
RETORNO	Fecha de inicio del cuatrimestre
<i>obtener_fin_cuatrimestre(Date fecha)</i>	
DESCRIPCIÓN	Obtiene la fecha de fin del cuatrimestre a partir de una fecha
ARGUMENTOS	Fecha en formato YYYY-mm-dd
FUNCIONAMIENTO	Obtiene el cuatrimestre en función de la fecha dada y los periodos determinados en las opciones de configuración y determina el día de fin de ese cuatrimestre
RETORNO	Fecha de fin del cuatrimestre

Tabla 95: Clase Create_Schedule

Clase Gest_Classroom	
<i>administrar_aulas()</i>	
DESCRIPCIÓN	Conforma toda la vista de la página de gestión de aulas
ARGUMENTOS	-
FUNCIONAMIENTO	Hace diversas llamadas al resto de métodos de la clase para conformar la vista de la página
RETORNO	-
<i>load_javascript_aula()</i>	
DESCRIPCIÓN	Genera el código javascript necesario para componer la vista
ARGUMENTOS	-
FUNCIONAMIENTO	Hace las llamadas a funciones necesarias y compone el código javascript de la página
RETORNO	-
<i>comprobar_peticion_aula()</i>	
DESCRIPCIÓN	Inserta un nuevo aula en la base de datos
ARGUMENTOS	-
FUNCIONAMIENTO	Comprueba si se ha solicitado la creación de un nuevo aula y la crea la entrada en la base de datos si es necesario
RETORNO	-

<i>load_html_aula()</i>	
DESCRIPCIÓN	Genera el código HTML del formulario de gestión de aulas
ARGUMENTOS	-
FUNCIONAMIENTO	Comprueba el proceso de creación y genera las distintas páginas en consecuencia
RETORNO	-
<i>alertas_aula()</i>	
DESCRIPCIÓN	Muestra un aviso cuando se inserta un nuevo aula
ARGUMENTOS	-
FUNCIONAMIENTO	Comprueba si se ha insertado un nuevo aula y muestra un mensaje en pantalla
RETORNO	-

Tabla 96: Clase Gest_Classroom

Clase Gest_Course	
<i>administrar_asignaturas()</i>	
DESCRIPCIÓN	Conforma toda la vista de la página de gestión de asignaturas
ARGUMENTOS	-
FUNCIONAMIENTO	Hace diversas llamadas al resto de métodos de la clase para conformar la vista de la página
RETORNO	-
<i>load_javascript_asignatura()</i>	
DESCRIPCIÓN	Genera el código javascript necesario para componer la vista
ARGUMENTOS	-
FUNCIONAMIENTO	Hace las llamadas a funciones necesarias y compone el código javascript de la página
RETORNO	-
<i>comprobar_peticion_asignatura()</i>	
DESCRIPCIÓN	Inserta una nueva asignatura en la base de datos
ARGUMENTOS	-
FUNCIONAMIENTO	Comprueba si se ha solicitado la creación de una nueva asignatura y crea la entrada en la base de datos si es necesario
RETORNO	-
<i>load_html_asignatura()</i>	
DESCRIPCIÓN	Genera el código HTML del formulario de gestión de asignaturas
ARGUMENTOS	-
FUNCIONAMIENTO	Comprueba el proceso de creación y genera las distintas páginas en consecuencia
RETORNO	-
<i>alertas_asignatura()</i>	
DESCRIPCIÓN	Muestra un aviso cuando se inserta una nueva asignatura
ARGUMENTOS	-
FUNCIONAMIENTO	Comprueba si se ha insertado una nueva asignatura y muestra un mensaje en pantalla
RETORNO	-

<i>generar_seleccion_grados()</i>	
DESCRIPCIÓN	Genera el código HTML de un desplegable con todos los distintos grados
ARGUMENTOS	-
FUNCIONAMIENTO	Recupera los grados de la base de datos y compone el código del desplegable en HTML
RETORNO	Código HTML del desplegable
<i>generar_seleccion_software()</i>	
DESCRIPCIÓN	Genera el código HTML de un desplegable con todos los distintos grados
ARGUMENTOS	-
FUNCIONAMIENTO	Recupera el software de la base de datos y compone el código del desplegable en HTML
RETORNO	Código HTML del desplegable
<i>generar_opcion_software()</i>	
DESCRIPCIÓN	Genera el código para el formulario de adhesión de software
ARGUMENTOS	-
FUNCIONAMIENTO	Recupera los grados de la base de datos y compone el código del desplegable
RETORNO	Código del desplegable
<i>generar_seleccion_maquinas_virtuales()</i>	
DESCRIPCIÓN	Genera el código HTML de un desplegable con todas las distintas máquinas virtuales
ARGUMENTOS	-
FUNCIONAMIENTO	Recupera las máquinas virtuales de la base de datos y compone el código del desplegable en HTML
RETORNO	Código HTML del desplegable
<i>generar_opcion_mv()</i>	
DESCRIPCIÓN	Genera el código para el formulario de adhesión de máquinas virtuales
ARGUMENTOS	-
FUNCIONAMIENTO	Recupera los grados de la base de datos y compone el código del desplegable
RETORNO	Código del desplegable

Tabla 97: Clase Gest_Course

Clase Gest_Degree	
<i>administrar_grados()</i>	
DESCRIPCIÓN	Conforma toda la vista de la página de gestión de grados
ARGUMENTOS	-
FUNCIONAMIENTO	Hace diversas llamadas al resto de métodos de la clase para conformar la vista de la página
RETORNO	-
<i>load_javascript_grado()</i>	
DESCRIPCIÓN	Genera el código javascript necesario para componer la vista
ARGUMENTOS	-
FUNCIONAMIENTO	Hace las llamadas a funciones necesarias y compone el código javascript de la página
RETORNO	-

<i>comprobar_peticion_grado()</i>	
DESCRIPCIÓN	Inserta un nuevo grado en la base de datos
ARGUMENTOS	-
FUNCIONAMIENTO	Comprueba si se ha solicitado la creación de un nuevo grado y la crea la entrada en la base de datos si es necesario
RETORNO	-
<i>load_html_grado()</i>	
DESCRIPCIÓN	Genera el código HTML del formulario de gestión de grados
ARGUMENTOS	-
FUNCIONAMIENTO	Comprueba el proceso de creación y genera las distintas páginas en consecuencia
RETORNO	-
<i>alertas_grado()</i>	
DESCRIPCIÓN	Muestra un aviso cuando se inserta un nuevo grado
ARGUMENTOS	-
FUNCIONAMIENTO	Comprueba si se ha insertado un nuevo grado y muestra un mensaje en pantalla
RETORNO	-

Tabla 98: Clase Gest_Degree

Clase Gest_Software	
<i>administrar_software()</i>	
DESCRIPCIÓN	Conforma toda la vista de la página de gestión de software
ARGUMENTOS	-
FUNCIONAMIENTO	Hace diversas llamadas al resto de métodos de la clase para conformar la vista de la página
RETORNO	-
<i>load_javascript_software()</i>	
DESCRIPCIÓN	Genera el código javascript necesario para componer la vista
ARGUMENTOS	-
FUNCIONAMIENTO	Hace las llamadas a funciones necesarias y compone el código javascript de la página
RETORNO	-
<i>comprobar_peticion_software()</i>	
DESCRIPCIÓN	Inserta un nuevo software en la base de datos
ARGUMENTOS	-
FUNCIONAMIENTO	Comprueba si se ha solicitado la creación de un nuevo software y crea la entrada en la base de datos si es necesario
RETORNO	-
<i>load_html_software()</i>	
DESCRIPCIÓN	Genera el código HTML del formulario de gestión de software
ARGUMENTOS	-
FUNCIONAMIENTO	Comprueba el proceso de creación y genera las distintas páginas en consecuencia
RETORNO	-

<i>alertas_software()</i>	
DESCRIPCIÓN	Muestra un aviso cuando se inserta un nuevo software
ARGUMENTOS	-
FUNCIONAMIENTO	Comprueba si se ha insertado un nuevo software y muestra un mensaje en pantalla
RETORNO	-

Tabla 99: Clase Gest_Software

Clase Gest_VM	
<i>administrar_maquinas_virtuales()</i>	
DESCRIPCIÓN	Conforma toda la vista de la página de gestión de máquinas virtuales
ARGUMENTOS	-
FUNCIONAMIENTO	Hace diversas llamadas al resto de métodos de la clase para conformar la vista de la página
RETORNO	-
<i>load_javascript_mv()</i>	
DESCRIPCIÓN	Genera el código javascript necesario para componer la vista
ARGUMENTOS	-
FUNCIONAMIENTO	Hace las llamadas a funciones necesarias y compone el código javascript de la página
RETORNO	-
<i>comprobar_peticion_mv()</i>	
DESCRIPCIÓN	Inserta una nueva máquina virtual en la base de datos
ARGUMENTOS	-
FUNCIONAMIENTO	Comprueba si se ha solicitado la creación de una nueva máquina virtual y la crea la entrada en la base de datos si es necesario
RETORNO	-
<i>load_html_mv()</i>	
DESCRIPCIÓN	Genera el código HTML del formulario de gestión de máquinas virtuales
ARGUMENTOS	-
FUNCIONAMIENTO	Comprueba el proceso de creación y genera las distintas páginas en consecuencia
RETORNO	-
<i>alertas_mv()</i>	
DESCRIPCIÓN	Muestra un aviso cuando se inserta una nueva máquina virtual
ARGUMENTOS	-
FUNCIONAMIENTO	Comprueba si se ha insertado una nueva máquina virtual y muestra un mensaje en pantalla
RETORNO	-

Tabla 100: Clase Gest_VM

Clase Gest_Data	
<i>administrar_informes()</i>	
DESCRIPCIÓN	Conforma toda la vista de la página de gestión de informes
ARGUMENTOS	-
FUNCIONAMIENTO	Hace diversas llamadas al resto de métodos de la clase para conformar la vista de la página
RETORNO	-

<i>load_html_data()</i>	
DESCRIPCIÓN	Genera el código HTML del formulario de gestión de estadísticas
ARGUMENTOS	-
FUNCIONAMIENTO	Comprueba el proceso de creación y genera las distintas páginas en consecuencia
RETORNO	-
<i>load_statistics(String cuatrimestre, int año)</i>	
DESCRIPCIÓN	Genera el código HTML de las estadísticas e informes para un cuatrimestre y año determinados
ARGUMENTOS	Cuatrimestre y año del que se desea obtener las estadísticas
FUNCIONAMIENTO	Recoge de la base de datos las reservas de un cuatrimestre y año y conforma el código HTML del informe
RETORNO	Código HTML del informe
<i>generate_excel(int año)</i>	
DESCRIPCIÓN	Genera el código del archivo Excel con los informes de ocupación para un año dado
ARGUMENTOS	Año del que se desean obtener las estadísticas
FUNCIONAMIENTO	Recoge de la base de datos las reservas de ese año y genera el código Excel del informe
RETORNO	Código Excel del informe
<i>printToday(int año)</i>	
DESCRIPCIÓN	Muestra un mensaje con el año de los informes que se muestran en pantalla
ARGUMENTOS	-
FUNCIONAMIENTO	Comprueba si se ha solicitado un año anterior y lo muestra, de lo contrario obtiene la fecha del día de hoy y la imprime en pantalla
RETORNO	Código HTML del mensaje
<i>generar_seleccion_annos(int año)</i>	
DESCRIPCIÓN	Muestra un desplegable con los años de los informes
ARGUMENTOS	-
FUNCIONAMIENTO	Comprueba los años en los que se han realizado reservas y genera un desplegable con dichos años
RETORNO	Código HTML del desplegable

Tabla 101: Clase Gest_Data

Clase Gest_Options	
<i>administrar_opciones()</i>	
DESCRIPCIÓN	Conforma toda la vista de la página de gestión de opciones
ARGUMENTOS	-
FUNCIONAMIENTO	Hace diversas llamadas al resto de métodos de la clase para conformar la vista de la página
RETORNO	-
<i>comprobar_peticion_opciones()</i>	
DESCRIPCIÓN	Actualiza las opciones de configuración
ARGUMENTOS	-
FUNCIONAMIENTO	Comprueba si se ha cambiado algún parámetro de configuración y actualiza los registros necesarios en la base de datos
RETORNO	-

<i>load_html_options()</i>	
DESCRIPCIÓN	Genera el código HTML del formulario de gestión de opciones
ARGUMENTOS	-
FUNCIONAMIENTO	Comprueba el proceso de creación y genera las distintas páginas en consecuencia
RETORNO	-
<i>alertas_opciones()</i>	
DESCRIPCIÓN	Muestra un aviso cuando se actualizan las opciones de configuración
ARGUMENTOS	-
FUNCIONAMIENTO	Comprueba si se ha modificado alguna configuración y muestra un mensaje en pantalla
RETORNO	-

Tabla 102: Clase Gest_Options

Clase Confirm_Schedule	
<i>confirmar_reservas</i>	
DESCRIPCIÓN	Conforma toda la vista de la página de confirmar reservas
ARGUMENTOS	-
FUNCIONAMIENTO	Hace diversas llamadas al resto de métodos de la clase para conformar la vista de la página
RETORNO	-
<i>comprobar_peticion_confirmacion ()</i>	
DESCRIPCIÓN	Comprueba si se ha confirmado una reserva y actualiza el registro en la BBDD
ARGUMENTOS	-
FUNCIONAMIENTO	Comprueba si se ha solicitado una confirmación de reserva y actualiza la entrada en la base de datos si es necesario
RETORNO	-
<i>load_html_confirmación()</i>	
DESCRIPCIÓN	Genera el código HTML del formulario de confirmar reservas
ARGUMENTOS	-
FUNCIONAMIENTO	Comprueba el proceso de confirmación y genera las distintas páginas en consecuencia
RETORNO	-

Tabla 103: Clase Confirm_Schedule

4.6. Diseño físico de datos

Abordamos ahora el **diseño físico de los datos**, es decir, el diseño de la base de datos, su esquema relacional y una descripción de cada uno de los campos almacenados:

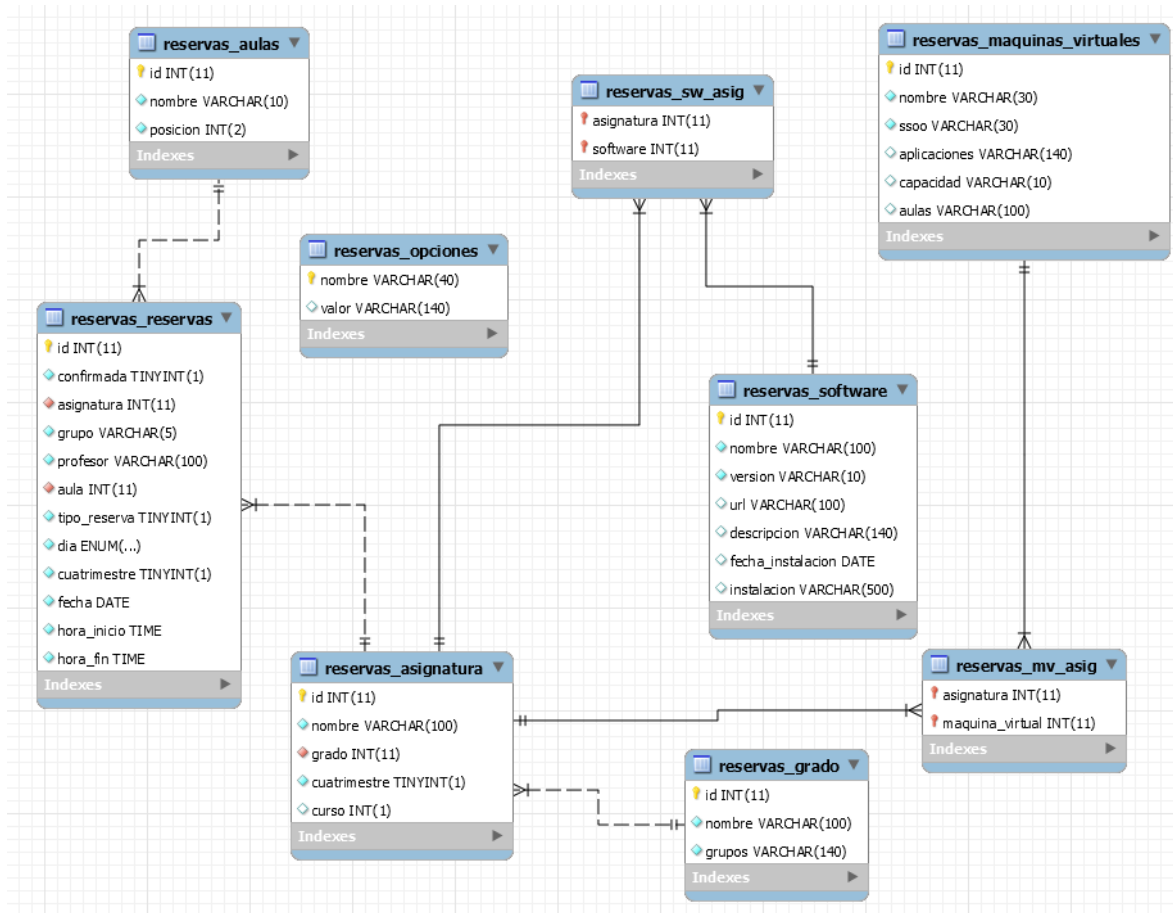


Figura 34: Esquema relacional

Tenemos un diseño con un total de 9 tablas donde cada una de ellas almacena los siguientes atributos:

Tabla reservas_opciones				
Atributo	Descripción	Tipo de Atributo	Tipo de Dato	Valores
NOMBRE	Identificador único del parámetro de configuración	Clave primaria	Varchar(40)	Nombres de opciones
VALOR	Valor almacenado del parámetro	-	Varchar(140)	Valor del parámetro de configuración

Tabla 104: Tabla reservas_opciones

Tabla reservas_aulas				
Atributo	Descripción	Tipo de Atributo	Tipo de Dato	Valores
ID	Identificador del aula	Clave primaria	int	0 a $2^{31} - 1$
NOMBRE	Nombre del aula	Atributo no nulo	Varchar(10)	Nombre del aula
POSICION	Orden del aula	Atributo no nulo	int	0 a $2^{31} - 1$

Tabla 105: Tabla reservas_aulas

Análisis, diseño e implantación de una aplicación para la gestión de reservas para las aulas docentes del departamento de informática

Tabla reservas_maquinas_virtuales				
Atributo	Descripción	Tipo de Atributo	Tipo de Dato	Valores
ID	Identificador único del parámetro de configuración	Clave primaria	int	0 a $2^{31} - 1$
NOMBRE	Nombre de la máquina virtual	Atributo no nulo	Varchar(30)	Nombre de la mv
Ssoo	Sistema Operativo instalado	Atributo no nulo	Varchar(30)	Sistema Operativo
APLICACIONES	Apps instaladas	-	Varchar(140)	Apps instaladas en la máquina virtual
CAPACIDAD	Memoria disponible	-	Varchar(10)	Capacidad del HDD de la máquina
AULAS	Aulas en las que está instalada la mv	-	Varchar(100)	Aulas en las que está instalada la máquina

Tabla 106: Tabla reservas_maquinas_virtuales

Tabla reservas_software				
Atributo	Descripción	Tipo de Atributo	Tipo de Dato	Valores
ID	Identificador del aula	Clave primaria	int	0 a $2^{31} - 1$
NOMBRE	Nombre del aula	Atributo no nulo	Varchar(100)	Nombre del aula
VERSION	Versión instalada	Atributo no nulo	Varchar(10)	Versión instalada
URL	Url de descarga	-	Varchar(100)	Enlace de descarga
DESCRIPCION	Información del software	-	Varchar(140)	Descripción del software instalado
FECHA_INSTALACION	Fecha de instalación	-	Date	Fecha de instalación
INSTALACION	Instrucciones de instalación	-	Varchar(500)	Instrucciones para la instalación

Tabla 107: Tabla reservas_software

Tabla reservas_grado				
Atributo	Descripción	Tipo de Atributo	Tipo de Dato	Valores
ID	Identificador único del parámetro de configuración	Clave primaria	int	0 a $2^{31} - 1$
NOMBRE	Nombre del grado	Atributo no nulo	Varchar(100)	Nombre del grado
GRUPOS	Grupos del grado	Atributo no nulo	Varchar(140)	Grupos del grado

Tabla 108: Tabla reservas_grado

Tabla reservas_asignatura				
Atributo	Descripción	Tipo de Atributo	Tipo de Dato	Valores
ID	Identificador del aula	Clave primaria	int	0 a $2^{31} - 1$
NOMBRE	Nombre del aula	Atributo no nulo	Varchar(100)	Nombre del aula
GRADO	Grado en que se cursa	Atributo no nulo	Int	Id del grado en que se cursa
CUATRIMESTRE	Cuatrimestre de la asignatura	Atributo no nulo	Byte	Número de cuatrimestre
CURSO	Curso de la asignatura	-	Int	Curso en que se imparte

Tabla 109: Tabla reservas_asignatura

Tabla reservas_mv_asignatura				
Atributo	Descripción	Tipo de Atributo	Tipo de Dato	Valores
ASIGNATURA	Id de asignatura	Clave primaria	int	0 a $2^{31} - 1$
MAQUINA_VIRTUAL	Id de máquina	Clave primaria	Int	0 a $2^{31} - 1$

Tabla 110: Tabla reservas_mv_asignatura

Tabla reservas_sw_asignatura				
Atributo	Descripción	Tipo de Atributo	Tipo de Dato	Valores
ASIGNATURA	Id de asignatura	Clave primaria	int	0 a $2^{31} - 1$
SOFTWARE	Id de software	Clave primaria	Int	0 a $2^{31} - 1$

Tabla 111: Tabla reservas_sw_asignatura

Tabla reservas_reservas				
Atributo	Descripción	Tipo de Atributo	Tipo de Dato	Valores
ID	Identificador de la reserva	Clave primaria	int	0 a $2^{31} - 1$
CONFIRMADA	Estado de la reserva	Atributo no nulo	Byte	0 o 1
ASIGNATURA	Id de asignatura	Clave ajena	int	0 a $2^{31} - 1$
GRUPO	Grupo de la reserva	Atributo no nulo	Varchar(5)	Número de grupo
PROFESOR	Profesor de la clase	Atributo no nulo	Varchar(100)	Correo electrónico del profesor
AULA	Id de aula	Clave ajena	int	0 a $2^{31} - 1$
TIPO_RESERVA	Tipo de reserva	Atributo no nulo	Byte	0 o 1
DIA	Día de la reserva	Atributo no nulo	Enumerado	Lunes, Martes, etc.
FECHA	Fecha de la reserva	Atributo no nulo	Date	Fecha de la reserva
HORA_INICIO	Hora de inicio	Atributo no nulo	Time	Hora de inicio
HORA_FIN	Hora de fin	Atributo no nulo	Time	Hora de fin

Tabla 112: Tabla reservas_reservas

5. Implementación e Implantación

El proceso de **implementación e implantación** tiene como objetivo la planificación de la entrega y aceptación final del sistema, y la realización de todas las actividades necesarias para garantizar su implantación.

5.1. Establecimiento del plan de implantación

La elección de una correcta estrategia de implantación beneficia directamente al sistema permitiendo anticiparse y calcular los recursos y el esfuerzo necesarios para mantener en funcionamiento el software diseñado. Para este caso particular, determinamos que el **plan de implantación** del sistema ha de seguir las siguientes fases:

- **Formación** tanto de los usuarios finales del sistema como del equipo encargado de definir las pruebas de implantación y aceptación.
- **Preparación** de la infraestructura para la incorporación del sistema al entorno de operación.
- **Instalación** de los componentes y realización de los procedimientos necesarios para la incorporación del sistema al entorno de operación.
- **Carga inicial** de los datos.
- **Realización de las pruebas** de implantación y aceptación.
- Establecimiento de un **plan de mantenimiento**.

Del mismo modo, es necesario definir una serie de **roles a desempeñar** durante todo el proceso de implantación del sistema, si bien varios roles pueden ser realmente desempeñados por una misma persona física:

- **Rol [1] – Responsable de implantación:** será el encargado de coordinar y supervisar la correcta implantación del sistema. Realizará un seguimiento de todo el proceso y ofrecerá soporte al resto de miembros.
- **Rol [2] – Administrador de base de datos:** encargado de gestionar la integridad y consistencia de la base de datos, así como de la carga inicial de los mismos.
- **Rol [3] – Administrador del servidor:** encargado del mantenimiento y gestión del servidor sobre el que correrá la aplicación final.
- **Rol [4] – Administrador del sitio web:** encargado de la gestión, seguridad y mantenimiento del sitio web sobre el que se construye la aplicación.

5.2. Formación necesaria

Cada miembro del equipo que desempeñe un rol durante la implantación deberá tener unos conocimientos y formación previa adecuados para el correcto desempeño de sus funciones que garantice el éxito del proceso. En función de los distintos perfiles y niveles de responsabilidad, y en caso de que fuese necesario, se proponen los siguientes **planes de formación**:

- Curso de 30 horas – MySQL: **principios de las BBDD relacionales**.
- Curso de 30 horas – Apache: **de lo básico a la Integración Software**.
- Curso de 15 horas – Wordpress: **aprenda Wordpress fácilmente**.

Como es lógico, los roles [2], [3] y [4] deberían realizar los cursos 1, 2 y 3 respectivamente, mientras que el rol de *Responsable de implantación* debe tener conocimientos de todas las áreas, esto es, debería realizar todos los cursos. Esto **garantizaría la formación mínima necesaria** para el proceso de implantación, aunque en el caso concreto de los técnicos de laboratorio que realizarán la implantación final su formación ya supera a la proporcionada en estos cursos, por lo que realmente no sería necesaria su realización para una implantación exitosa del sistema.

En el caso de los usuarios finales de la aplicación podemos distinguir entre los distintos tipos de usuario que ya vimos al principio de este documento. De esta forma, los técnicos de laboratorio deberían al menos leer el manual de usuario del sistema mientras que en el caso de profesores y alumnos no es necesaria una formación previa para las funciones que desempeñarán con la aplicación (aunque podría ser recomendable en algunos casos).

5.3. Elaboración de los manuales

Se han redactado una buena cantidad de documentación sobre el funcionamiento del sistema en forma de manuales destinados a cada uno de los roles finales que harán uso de la aplicación. Dichos manuales están incluidos en el presente documento como apéndices, y son los siguientes:

- **Manual de administrador.** Contiene información adicional acerca de cómo realizar las gestiones de administración.

5.4. Incorporación del sistema al entorno de operación

El objetivo de esta tarea es realizar la instalación del sistema base para la incorporación posterior de los componentes asociados a los sistemas de información implicados en la implantación. El **sistema base** consiste en los siguientes elementos:

- **Equipo Servidor:** Servidor web Apache HTTP Server, Sistema Operativo Debian 7.8 Jessie y el Sistema Gestor de Bases de Datos MySQL.

La **instalación de WordPress** por su parte requiere la realización de un paso previo para su instalación junto con el tema Shape5 Vertex Response, que será utilizado como plantilla base para el css del sitio web. Los **pasos a seguir** son los siguientes:

1. Descomprimir el archivo *wordpress-4.1.1-es_ES.zip* en la carpeta correspondiente al servidor web (típicamente la ruta */var/www/*).
2. Descomprimir el archivo *SiteShaper_Shape5_Vertex_WP.zip* en la misma carpeta.
3. Realizar el proceso típico de instalación de WordPress.

Una vez realizada la instalación y configuración básica de WordPress además de la construcción de las distintas secciones de las que consta la página web del LDI, el siguiente paso es la **instalación del plugin Gestor de Reservas** desarrollado que incluirá las funciones de administración de aulas deseadas. Para ello debe descomprimirse el archivo *reservas.rar* en *wordpress/wp-content/plugins/*, y posteriormente activar el plugin desde las opciones de administración de plugins de WordPress.

En su primer inicio del plugin creará automáticamente la base de datos a partir del esquema básico elaborado en el proceso de análisis y añadirá las opciones de configuración predeterminadas al sistema. Hecho esto el último paso consistiría en crear las páginas adicionales que muestren la información a los usuarios y cuyo funcionamiento puede verse en el *Manual de Administración* incluido en los apéndices de este documento.

5.5. Carga de datos al entorno de operación

No existe, a excepción de los parámetros de configuración predeterminados, **ninguna carga inicial de datos** que deba añadirse durante la instalación del sistema, ya que toda la información correspondiente a las reservas debe ser

añadida de forma manual por los técnicos del laboratorio que gestionarán la aplicación y supervisarán la correcta incorporación de cada una de las reservas al sistema.

La carga de datos pues debe realizarse preferiblemente a través de la propia interfaz de la aplicación que le guiará durante el proceso de creación de cada nuevo elemento de datos.

5.6. Preparación del mantenimiento del sistema

En lo referente al **mantenimiento del sistema**, se propone la utilización de dos entornos bien diferenciados que permitan garantizar la continuidad y el correcto funcionamiento del sistema al mismo tiempo que agiliza la tarea de corrección de errores e inclusión de nuevas características en caso de que fuese necesario.

Entorno de pruebas

En el entorno de pruebas se realizarán las **primeras correcciones** cuando surja un problema o se realice una actualización de software de algunos de los componentes externos utilizados en el sistema. Es especialmente importante realizar un control exhaustivo sobre cada una de las nuevas versiones de WordPress a fin de garantizar el correcto funcionamiento del componente desarrollado con cada nueva versión del CMS.

Este entorno dispondrá de todas las funcionalidades que compone la aplicación para poder probar los nuevos cambios integrados con el resto de módulos que forman el sistema.

Una vez se superen las pruebas, las modificaciones se pasarán al entorno de producción.

Entorno de producción

Es el entorno real sobre el que funciona la aplicación. Se realizará en todo momento un **seguimiento de las posibles modificaciones** o actualizaciones realizadas hasta asegurarse que todo cambio en el sistema es efectivo y estable.

El entorno de producción solo debe actualizarse una vez se hayan superado satisfactoriamente todas las pruebas en el entorno de pruebas.

El desarrollador no se compromete a impartir cursos de formación al equipo de mantenimiento salvo en las cuestiones estrictamente referentes al funcionamiento del módulo *Gestor de Reservas*. Si se ofrecerá, no obstante, un soporte habilitado con la finalidad de **resolver dudas al equipo de mantenimiento** a través de contacto telefónico o personal en caso de que fuese estrictamente necesario. La duración de dicho soporte tendrá una extensión de tres meses desde la finalización del proyecto. Una vez transcurrido el tiempo de soporte no se garantizará la asistencia por parte del desarrollador.

Por otro lado, cualquier error o inconsistencia que sufra la aplicación durante el periodo de adaptación a causa de un mal diseño del sistema, será solventado por el desarrollador de manera totalmente gratuita.

5.7. Presentación y aprobación del sistema

Una vez superadas todas las pruebas de implantación y aceptación, se presentará el sistema al Laboratorio del Departamento de Informática con el fin de formalizar la aprobación del sistema.

Se acordará así una **reunión con el cliente** para exponer cada punto del sistema y se realizará una presentación del mismo. En esa reunión es indispensable la presencia tanto del desarrollador, Francisco Javier Rodríguez Isabel, como la de los tres técnicos del LDI: Óscar Pérez Alonso, Jaime Pons Bailly-Bailliere y Roberto Fuentes Astorga.

Existirá, a partir de ese momento, un **periodo de prueba de 30 días naturales** cuyo fin es permitir al cliente comprobar la correcta funcionalidad de la herramienta, tras el cual deberá indicar al desarrollador si se decide a aprobar el sistema.

5.8. Paso a producción

El paso a producción se llevará a cabo de forma planificada, y será el momento en el que se **traspasa la responsabilidad al equipo de mantenimiento del LDI**. Este proceso se llevará a cabo en dos pasos bien diferenciados:

1. **Análisis y determinación de los componentes** a incorporar en producción. Se seleccionarán las partes del Entorno de pruebas que deben ser llevadas al entorno de producción. Tras este estudio se realizará la instalación de los distintos componentes y se cargarán los datos iniciales si los hubiera.

Se comprobará también la correcta instalación del sistema y se determinará la fecha de activación del mismo así como la eliminación del antiguo sistema en caso de existir. Se establecerá como se llevará a cabo la transición de uno a otro.

2. Una vez probada la correcta instalación del sistema y su correcto funcionamiento, se comunicará al cliente la **total disponibilidad del sistema** para su uso. A partir de ese momento comenzará el servicio de mantenimiento bajo las condiciones anteriormente indicadas.

6. Evaluación

En este apartado se detalla la metodología seguida para comprobar que el sistema cumple con los objetivos deseados, tanto a nivel técnico (garantizando el correcto funcionamiento a través del Plan de Pruebas) como a nivel personal a través de la realización de un test a los diferentes puntos de interés.

6.1. Plan de Pruebas

El establecimiento de un **Plan de Pruebas completo y riguroso** es vital para conseguir la mayor calidad posible durante el desarrollo del software. Así pues, se propone el siguiente catálogo de pruebas a realizar categorizadas además en diferentes tipos: pruebas unitarias, de integración, del sistema, de implantación del sistema y de aceptación del sistema. Para que la aplicación pase a producción será requisito indispensable superar todas y cada una de estas pruebas.

Pruebas unitarias

Con las **pruebas unitarias** podemos asegurar el correcto funcionamiento de fragmentos de código. Además las pruebas unitarias se dividen en dos tipos diferentes:

- **Pruebas de caja blanca.** Se centran en los detalles del software y están estrechamente relacionadas con el código fuente del programa.
- **Pruebas de caja negra.** Cuando solo se estudia la entrada y salida del sistema del que se realizan las pruebas.

Definiremos estas pruebas en tablas, donde cada prueba llevará asociada un identificador del tipo PU-Xnnn, donde cada letra indica lo siguiente:

- PU: indica que se trata de una prueba unitaria.
- X: determinará si se trata de pruebas de caja blanca (B) o si son pruebas de caja negra (N).
- nnn: indica el número de prueba, con un valor entre 1 y 999.

No obstante, las pruebas de caja blanca son más complejas y costosas de realizar, aunque son muy recomendables cuando la complejidad Ciclomática de los métodos es 2 o mayor. Nuestra aplicación tiene una complejidad Ciclomática simple, por lo que se ha tomado la decisión de ahorrar recursos y **no realizar las pruebas de caja blanca.**

Para las **pruebas de caja negra**, por otra parte, se propone el siguiente catálogo para probar los métodos más importantes en cada clase:

PU-N001	
Descripción	Comprobar el correcto funcionamiento del método que comprueba las llamadas para mostrar las vistas de reservas por día o aula mediante el uso de los <i>shortcodes</i> de <i>WordPress</i> . La función debe ser invocada cuando se escribe el <i>shortcode</i> en una página y debe mostrar la información solicitada sobre las reservas programadas.
Módulo	Vista
Clase	Calendar
Método	shortcode(atts[])
Entrada	[reservas dia=Lunes] (<i>shortcode</i>)
Salida válida	Calendario con las reservas programadas para el Lunes

Tabla 113: Prueba PU-N001

PU-N002	
Descripción	Comprobar el correcto funcionamiento del método que comprueba las llamadas para mostrar el panel de usuarios que permite a los usuarios iniciar sesión en la aplicación para poder solicitar una nueva reserva. La función debe ser invocada cuando se escribe el <i>shortcode</i> en una página y debe mostrar el panel de inicio de sesión.
Módulo	Vista
Clase	User
Método	shortcode_login(atts[])
Entrada	[login] (<i>shortcode</i>)
Salida válida	Panel de inicio de sesión

Tabla 114: Prueba PU-N002

PU-N003	
Descripción	Comprobar el correcto funcionamiento del método que registra el panel del Gestor de Reservas en el menú de administración de Wordpress
Módulo	Controlador
Clase	Administrator
Método	reg_panel_admin()
Entrada	-
Salida válida	Panel de reservas añadido en el menú de administración de Wordpress

Tabla 115: Prueba PU-N003

PU-N004	
Descripción	Comprobar el correcto funcionamiento del método que compone la página de gestión de reservas
Módulo	Vista
Clase	Gest_Schedule
Método	administrar_reservas()
Entrada	-
Salida válida	Página de gestión de reservas

Tabla 116: Prueba PU-N004

PU-N005	
Descripción	Comprobar el correcto funcionamiento del método que compone la página de gestión de aulas
Módulo	Vista
Clase	Gest_Classroom
Método	administrar_aulas()
Entrada	-
Salida válida	Página de gestión de aulas

Tabla 117: Prueba PU-N005

PU-N006	
Descripción	Comprobar el correcto funcionamiento del método que compone la página de gestión de asignaturas
Módulo	Vista
Clase	Gest_Course
Método	administrar_asignaturas()
Entrada	-
Salida válida	Página de gestión de asignaturas

Tabla 118: Prueba PU-N006

PU-N007	
Descripción	Comprobar el correcto funcionamiento del método que compone la página de gestión de grados
Módulo	Vista
Clase	Gest_Degree
Método	administrar_grados()
Entrada	-
Salida válida	Página de gestión de grados

Tabla 119: Prueba PU-N007

PU-N008	
Descripción	Comprobar el correcto funcionamiento del método que compone la página de gestión de software
Módulo	Vista
Clase	Gest_Software
Método	administrar_software()
Entrada	-
Salida válida	Página de gestión de software

Tabla 120: Prueba PU-N008

PU-N009	
Descripción	Comprobar el correcto funcionamiento del método que compone la página de gestión de máquinas virtuales
Módulo	Vista
Clase	Gest_MV
Método	administrar_mv()
Entrada	-
Salida válida	Página de gestión de máquinas virtuales

Tabla 121: Prueba PU-N009

PU-N010	
Descripción	Comprobar el correcto funcionamiento del método que compone la página de opciones
Módulo	Vista
Clase	Gest_Options
Método	administrar_opciones()
Entrada	-
Salida válida	Página de opciones

Tabla 122: Prueba PU-N010

PU-N011	
Descripción	Comprobar el correcto funcionamiento del método que compone la página de gestión de informes
Módulo	Vista
Clase	Gest_Data
Método	administrar_informes()
Entrada	-
Salida válida	Página de gestión de informes

Tabla 123: Prueba PU-N011

PU-N012	
Descripción	Comprobar el correcto funcionamiento del método que define las tablas en la base de datos
Módulo	Modelo
Clase	Database
Método	definirTablas()
Entrada	-
Salida válida	Estructura de datos creada en la base de datos

Tabla 124: Prueba PU-N012

PU-N013	
Descripción	Comprobar el correcto funcionamiento del método que recupera datos de la tabla de opciones
Módulo	Modelo
Clase	Database
Método	get_opcion(nombre)
Entrada	Nombre de la opción que se desea recuperar
Salida válida	Valor de la opción solicitada

Tabla 125: Prueba PU-N013

PU-N014	
Descripción	Comprobar el correcto funcionamiento del método que recupera datos de la tabla de reservas
Módulo	Modelo
Clase	Database
Método	get_reservas(atributo, valor)
Entrada	Nombre del atributo y valor por el que se filtrará la búsqueda
Salida válida	Lista de reservas que coinciden con los parámetros de búsqueda

Tabla 126: Prueba PU-N014

PU-N015	
Descripción	Comprobar el correcto funcionamiento del método que recupera las reservas pendientes de la base de datos
Módulo	Modelo
Clase	Database
Método	get_reservas_pendientes()
Entrada	-
Salida válida	Todas las reservas pendientes almacenadas en la base de datos

Tabla 127: Prueba PU-N015

PU-N016	
Descripción	Comprobar el correcto funcionamiento del método que inserta una reserva en la base de datos
Módulo	Modelo
Clase	Database
Método	set_reserva(reserva)
Entrada	Atributos de la reserva que se quiere almacenar
Salida válida	Nuevo objeto insertado en la base de datos

Tabla 128: Prueba PU-N016

PU-N017	
Descripción	Comprobar el correcto funcionamiento del método que recupera aulas de la base de datos
Módulo	Modelo
Clase	Database
Método	get_aulas(atributo,valor)
Entrada	Nombre del atributo y valor por el que se filtrará la búsqueda
Salida válida	Lista de las aulas que coinciden con los parámetros de búsqueda

Tabla 129: Prueba PU-N017

PU-N018	
Descripción	Comprobar el correcto funcionamiento del método que inserta un aula en la base de datos
Módulo	Modelo
Clase	Database
Método	set_aula(aula)
Entrada	Atributos del aula que se quiere almacenar
Salida válida	Nuevo objeto insertado en la base de datos

Tabla 130: Prueba PU-N018

PU-N019	
Descripción	Comprobar el correcto funcionamiento del método que recupera asignaturas de la base de datos
Módulo	Modelo
Clase	Database
Método	get_asignaturas(atributo,valor)
Entrada	Nombre del atributo y valor por el que se filtrará la búsqueda
Salida válida	Lista de las asignaturas que coinciden con los parámetros de búsqueda

Tabla 131: Prueba PU-N019

PU-N020	
Descripción	Comprobar el correcto funcionamiento del método que inserta una asignatura en la base de datos
Módulo	Modelo
Clase	Database
Método	set_asignatura(asignatura)
Entrada	Atributos de la asignatura que se quiere almacenar
Salida válida	Nuevo objeto insertado en la base de datos

Tabla 132: Prueba PU-N020

PU-N021	
Descripción	Comprobar el correcto funcionamiento del método que recupera grados de la base de datos
Módulo	Modelo
Clase	Database
Método	get_grados(atributo,valor)
Entrada	Nombre del atributo y valor por el que se filtrará la búsqueda
Salida válida	Lista de las asignaturas que coinciden con los parámetros de búsqueda

Tabla 133: Prueba PU-N021

PU-N022	
Descripción	Comprobar el correcto funcionamiento del método que inserta un grado en la base de datos
Módulo	Modelo
Clase	Database
Método	set_grado(grado)
Entrada	Atributos del grado que se quiere almacenar
Salida válida	Nuevo objeto insertado en la base de datos

Tabla 134: Prueba PU-N022

PU-N023	
Descripción	Comprobar el correcto funcionamiento del método que recupera software de la base de datos
Módulo	Modelo
Clase	Database
Método	get_software(atributo,valor)
Entrada	Nombre del atributo y valor por el que se filtrará la búsqueda
Salida válida	Lista de las asignaturas que coinciden con los parámetros de búsqueda

Tabla 135: Prueba PU-N023

PU-N024	
Descripción	Comprobar el correcto funcionamiento del método que inserta un software en la base de datos
Módulo	Modelo
Clase	Database
Método	set_software(software)
Entrada	Atributos del software que se quiere almacenar
Salida válida	Nuevo objeto insertado en la base de datos

Tabla 136: Prueba PU-N024

PU-N025	
Descripción	Comprobar el correcto funcionamiento del método que recupera máquinas virtuales de la base de datos
Módulo	Modelo
Clase	Database
Método	get_maquinas_virtuales(atributo,valor)
Entrada	Nombre del atributo y valor por el que se filtrará la búsqueda
Salida válida	Lista de las asignaturas que coinciden con los parámetros de búsqueda

Tabla 137: Prueba PU-N025

PU-N026	
Descripción	Comprobar el correcto funcionamiento del método que inserta una máquina virtual en la base de datos
Módulo	Modelo
Clase	Database
Método	set_maquina_virtual(maquina_virtual)
Entrada	Atributos de la máquina virtual que se quiere almacenar
Salida válida	Nuevo objeto insertado en la base de datos

Tabla 138: Prueba PU-N026

PU-N027	
Descripción	Comprobar el correcto funcionamiento del método que proporciona el nombre de un profesor a partir de su correo electrónico. Para ello realiza una consulta al LDAP de la universidad
Módulo	Modelo
Clase	Database
Método	get_profesor(correo)
Entrada	Correo electrónico del profesor
Salida válida	Nombre y apellidos del profesor

Tabla 139: Prueba PU-N027

PU-N028	
Descripción	Comprobar el correcto funcionamiento del método que obtiene el correo electrónico de todos los profesores que tienen alguna reserva asignada en el sistema
Módulo	Modelo
Clase	Database
Método	get_profesores()
Entrada	-
Salida válida	Lista con los correos electrónicos de los profesores

Tabla 140: Prueba PU-N028

PU-N029	
Descripción	Comprobar el correcto funcionamiento del método que proporciona los datos de ocupación de las aulas para los informes
Módulo	Modelo
Clase	Database
Método	get_reservas_for_statistics(cuatrimestre,año)
Entrada	Cuatrimestre y año del que se desean los datos
Salida válida	Lista con las reservas que coinciden con los parámetros de búsqueda

Tabla 141: Prueba PU-N029

Pruebas de integración

Las **pruebas de integración** nos permiten probar que los distintos procesos de la aplicación funcionan correctamente. De nuevo vamos a representar las pruebas en tablas, donde cada una contará con un identificador del tipo PI-nnn, donde nnn es el número de prueba:

PI-001	
Descripción	Ir a la página de ocupación de aulas de un día determinado
Clase	Calendar
Entrada	Sin datos.
Secuencia de pasos	<ol style="list-style-type: none"> 1. Entrar en la web del Laboratorio del Departamento de Informática 2. Acceder a una de las páginas que muestran la ocupación diaria
Salida	Aparecerá un pequeño cronograma con las distintas reservas programadas
Precondiciones	La página web debe estar accesible desde el navegador Haber introducido reservas para ese día en algún aula

Tabla 142: Prueba PI-001

PI-002	
Descripción	Ir a la página de ocupación de aulas de un aula determinada
Clase	Calendar
Entrada	Sin datos.
Secuencia de pasos	<ol style="list-style-type: none"> 1. Entrar en la web del Laboratorio del Departamento de Informática 2. Acceder a la página que muestra la ocupación del aula
Salida	Aparecerá un pequeño cronograma con las distintas reservas programadas
Precondiciones	La página web debe ser accesible desde el navegador Haber introducido un aula en el sistema Haber introducido reservas para el aula

Tabla 143: Prueba PI-002

PI-003	
Descripción	Acceder al panel de solicitud de reservas
Clase	User
Entrada	Datos de inicio de sesión (correo y contraseña)
Secuencia de pasos	<ol style="list-style-type: none"> 1. Entrar en la web del Laboratorio del Departamento de Informática 2. Acceder a la página de solicitud de reservas 3. Introducir los credenciales de inicio de sesión
Salida	Acceso al panel de solicitud de reservas
Precondiciones	La página web debe estar accesible desde el navegador La conexión con el servidor LDAP debe estar configurada

Tabla 144: Prueba PI-003

PI-004	
Descripción	Realizar la solicitud de una reserva
Clase	User
Entrada	Datos de la reserva
Secuencia de pasos	<ol style="list-style-type: none"> 1. Entrar en la web del Laboratorio del Departamento de Informática 2. Acceder a la página de solicitud de reservas 3. Introducir los credenciales de inicio de sesión 4. Completar el formulario de solicitud de reserva
Salida	Se enviará un correo electrónico a la cuenta del LDI
Precondiciones	La página web debe ser accesible desde el navegador La conexión con el servidor LDAP debe estar configurada

Tabla 145: Prueba PI-004

PI-005	
Descripción	Confirmar una reserva pendiente
Clase	Confirm_Schedule
Entrada	Reserva solicitada
Secuencia de pasos	<ol style="list-style-type: none"> 1. Abrir el enlace enviado al correo electrónico 2. Iniciar sesión con una cuenta de administrador 3. Confirmar la reserva
Salida	La reserva será visible en el sistema
Precondiciones	Debe haberse solicitado una reserva

Tabla 146: Prueba PI-005

PI-006	
Descripción	Editar o borrar una reserva
Clase	Gest_Schedule
Entrada	Sin datos.
Secuencia de pasos	<ol style="list-style-type: none"> 1. Entrar en la web del Laboratorio del Departamento de Informática 2. Acceder al panel de administración 3. Ir a la página de gestión de reservas 4. Editar o borrar una reserva
Salida	Se eliminará la reserva de la base de datos
Precondiciones	La página web debe ser accesible desde el navegador Debe existir al menos una reserva en el sistema

Tabla 147: Prueba PI-006

PI-007	
Descripción	Añadir una nueva reserva
Clase	Create_Schedule
Entrada	Datos de la reserva
Secuencia de pasos	<ol style="list-style-type: none"> 1. Entrar en la web del Laboratorio del Departamento de Informática 2. Acceder al panel de administración 3. Ir a la página de creación de reservas 4. Completar el formulario
Salida	Se añade una nueva reserva en el sistema
Precondiciones	La página web debe ser accesible desde el navegador

Tabla 148: Prueba PI-007

PI-008	
Descripción	Añadir un nuevo aula
Clase	Gest_Classroom
Entrada	Datos del aula
Secuencia de pasos	<ol style="list-style-type: none"> 1. Entrar en la web del Laboratorio del Departamento de Informática 2. Acceder al panel de administración 3. Ir a la página de gestión de aulas 4. Completar el formulario para añadir un nuevo aula
Salida	Se añade un nuevo aula en el sistema
Precondiciones	La página web debe ser accesible desde el navegador

Tabla 149: Prueba PI-008

PI-009	
Descripción	Editar o borrar un aula
Clase	Gest_Classroom
Entrada	Sin datos.
Secuencia de pasos	<ol style="list-style-type: none"> 1. Entrar en la web del Laboratorio del Departamento de Informática 2. Acceder al panel de administración 3. Ir a la página de gestión de aulas 4. Editar o borrar un aula
Salida	Se modificará o eliminará el aula de la base de datos
Precondiciones	La página web debe ser accesible desde el navegador Debe existir al menos un aula en el sistema

Tabla 150: Prueba PI-009

PI-010	
Descripción	Añadir una nueva asignatura
Clase	Gest_Course
Entrada	Datos de la asignatura
Secuencia de pasos	<ol style="list-style-type: none"> 1. Entrar en la web del Laboratorio del Departamento de Informática 2. Acceder al panel de administración 3. Ir a la página de gestión de asignaturas 4. Completar el formulario para añadir una nueva asignatura
Salida	Se añade una nueva asignatura en el sistema
Precondiciones	La página web debe ser accesible desde el navegador

Tabla 151: Prueba PI-010

PI-011	
Descripción	Editar o borrar una asignatura
Clase	Gest_Course
Entrada	Sin datos.
Secuencia de pasos	<ol style="list-style-type: none"> 1. Entrar en la web del Laboratorio del Departamento de Informática 2. Acceder al panel de administración 3. Ir a la página de gestión de asignaturas 4. Editar o borrar una asignatura
Salida	Se modificará o eliminará la asignatura de la base de datos
Precondiciones	La página web debe ser accesible desde el navegador Debe existir al menos una asignatura en el sistema

Tabla 152: Prueba PI-011

PI-012	
Descripción	Añadir un nuevo grado
Clase	Gest_Degree
Entrada	Datos del grado
Secuencia de pasos	<ol style="list-style-type: none"> 1. Entrar en la web del Laboratorio del Departamento de Informática 2. Acceder al panel de administración 3. Ir a la página de gestión de grados 4. Completar el formulario para añadir un nuevo grado
Salida	Se añade un nuevo grado en el sistema
Precondiciones	La página web debe ser accesible desde el navegador

Tabla 153: Prueba PI-012

PI-013	
Descripción	Editar o borrar un grado
Clase	Gest_Degree
Entrada	Sin datos.
Secuencia de pasos	<ol style="list-style-type: none"> 1. Entrar en la web del Laboratorio del Departamento de Informática 2. Acceder al panel de administración 3. Ir a la página de gestión de grados 4. Editar o borrar un grado
Salida	Se modificará o eliminará el grado de la base de datos
Precondiciones	La página web debe ser accesible desde el navegador Debe existir al menos un grado en el sistema

Tabla 154: Prueba PI-013

PI-014	
Descripción	Añadir un nuevo software
Clase	Gest_Software
Entrada	Datos del software
Secuencia de pasos	<ol style="list-style-type: none"> 1. Entrar en la web del Laboratorio del Departamento de Informática 2. Acceder al panel de administración 3. Ir a la página de gestión de software 4. Completar el formulario para añadir un nuevo software
Salida	Se añade un nuevo software en el sistema
Precondiciones	La página web debe ser accesible desde el navegador

Tabla 155: Prueba PI-014

PI-015	
Descripción	Editar o borrar un software
Clase	Gest_Software
Entrada	Sin datos.
Secuencia de pasos	<ol style="list-style-type: none"> 1. Entrar en la web del Laboratorio del Departamento de Informática 2. Acceder al panel de administración 3. Ir a la página de gestión de software 4. Editar o borrar un software
Salida	Se modificará o eliminará el software de la base de datos
Precondiciones	La página web debe ser accesible desde el navegador Debe existir al menos un software en el sistema

Tabla 156: Prueba PI-015

PI-016	
Descripción	Añadir una nueva máquina virtual
Clase	Gest_MV
Entrada	Datos de la máquina virtual
Secuencia de pasos	<ol style="list-style-type: none"> 1. Entrar en la web del Laboratorio del Departamento de Informática 2. Acceder al panel de administración 3. Ir a la página de gestión de máquinas virtuales 4. Completar el formulario para añadir una nueva máquina virtual
Salida	Se añade una nueva máquina virtual en el sistema
Precondiciones	La página web debe ser accesible desde el navegador

Tabla 157: Prueba PI-016

PI-017	
Descripción	Descargar el informe de aulas
Clase	Gest_Data
Entrada	Sin datos.
Secuencia de pasos	<ol style="list-style-type: none"> 1. Entrar en la web del Laboratorio del Departamento de Informática 2. Acceder al panel de administración 3. Ir a la página de informes 4. Descargar el informe
Salida	Documento .xls con los informes
Precondiciones	La página web debe ser accesible desde el navegador

Tabla 158: Prueba PI-017

PI-018	
Descripción	Modificar las opciones de configuración
Clase	Gest_Options
Entrada	Nuevos datos para las opciones
Secuencia de pasos	<ol style="list-style-type: none"> 1. Entrar en la web del Laboratorio del Departamento de Informática 2. Acceder al panel de administración 3. Ir a la página de opciones 4. Cambiar algún parámetro de configuración
Salida	Se actualizan los parámetros en la base de datos
Precondiciones	La página web debe ser accesible desde el navegador

Tabla 159: Prueba PI-018

Pruebas de implantación del sistema

Las **pruebas de implantación** del sistema nos aseguran que el sistema se ha instalado correctamente y todas las partes son funcionales. Estas pruebas serán definidas con el identificador PIS-*nnn*, donde *nnn* es el número de prueba:

PIS-001	
Descripción	Comprobar que la web está instalada correctamente en el servidor.
Entrada	Se comprueba cuando se inicia el sistema.
Secuencia de pasos	<ol style="list-style-type: none"> 1. Se accede a la web del Laboratorio del Departamento de Informática.
Salida	La web funciona y muestra la información con normalidad
Precondiciones	Servidor iniciado, servidor de BBDD iniciado

Tabla 160: Prueba PIS-001

PIS-002	
Descripción	Comprobar que la aplicación está instalada correctamente en Wordpress.
Entrada	Se comprueba que se accede correctamente a la aplicación
Secuencia de pasos	<ol style="list-style-type: none"> 1. Se accede a la web del Laboratorio del Departamento de Informática. 2. Se accede al panel de administración 3. En la sección de plugin, se comprueba que "Reservas" está activo.
Salida	La aplicación funciona correctamente
Precondiciones	Servidor y BBDD iniciado, Wordpress instalado

Tabla 161: Prueba PIS-002

PIS-003	
Descripción	Comprobar la conexión con el servidor LDAP
Entrada	Se comprueba cuando se inicia el sistema.
Secuencia de pasos	<ol style="list-style-type: none"> 1. Se accede a la web del Laboratorio del Departamento de Informática. 2. Se comprueba que los nombres de los profesores de las reservas son correctos. 3. Se comprueba que se puede iniciar sesión para solicitar reserva
Salida	La web muestra el nombre completo de los profesores y permite a estos iniciar sesión
Precondiciones	Servidor iniciado, servidor de BBDD iniciado, servidor LDAP configurado e iniciado.

Tabla 162: Prueba PIS-003

Pruebas de aceptación del sistema

Las **pruebas de aceptación del sistema** tienen como objetivo comprobar si el producto software se puede clasificar como aceptable y si está listo para su paso a producción. Para nuestro caso particular se ha optado por hacer un test de satisfacción de usuarios, realizado a los clientes de la aplicación y cuyo contenido y resultados se detallan en el siguiente apartado.

6.2. Satisfacción de usuarios

Una vez garantizado el correcto funcionamiento a través del plan de pruebas de cada uno de los elementos del sistema, se realiza un test con el objetivo de **proporcionar al desarrollador un *feedback*** acerca del funcionamiento y la facilidad del sistema desarrollado. Así pues se tratará de averiguar si el *software* es usable, si existe margen de mejora en la interfaz de usuario o comprobar qué funcionalidades se podrían agregar al sistema. Se espera recoger también las impresiones generales de los usuarios al utilizar la aplicación.

Así pues, el test consiste en una serie de preguntas en las que se recogieron los siguientes resultados por parte del equipo de pruebas, que consistió en los 3 técnicos de laboratorio (uno de ellos profesor de la universidad) y los tres becarios del mismo:

1. *¿Qué le parece la nueva web del Laboratorio del Departamento de Informática?*

Respuesta 1: Una web sencilla y funcional que se adapta más a la nueva imagen de la web de la Universidad.

Respuesta 2: Bien, bonito diseño y funcional.

Respuesta 3: Me gusta que sea adaptativa.

Respuesta 4: Útil y sencilla de manejar.

Respuesta 5: Está muy bien, tiene un aspecto moderno y su navegación es intuitiva.

Respuesta 6: Organiza mejor la información y los contenidos que la antigua, además me parece una buena idea utilizar la misma estructura e interfaz que la del departamento.

2. *¿Mejoraría de algún modo la interfaz de la nueva web del Laboratorio del Departamento de Informática?*

Respuesta 1: Intentaría que el logo del laboratorio estuviera más integrado con la cabecera.

Respuesta 2: Algunas mejoras en el aspecto de la interfaz.

Respuesta 3: Algunas mejoras en el aspecto de la interfaz.

Respuesta 4: Algunas mejoras en el aspecto de la interfaz.

Respuesta 5: Algunas mejoras en el aspecto de la interfaz.

Respuesta 6: Algunas mejoras en el aspecto de la interfaz.

3. *Sobre la ocupación de las aulas. ¿Encontró fácilmente la información que buscaba?*

Respuesta 1: Muy fácil.

Respuesta 2: Sí.

Respuesta 3: Sí, está colocado perfectamente en el menú principal.

Respuesta 4: Sí, está fácilmente visible en el menú derecho.

Respuesta 5: Sí. La navegación es muy intuitiva.

Respuesta 6: Sí.

4. *Sobre la solicitud de una nueva reserva. ¿Pudo iniciar sesión sin problemas en la aplicación?*

Respuesta 1: Sí.

Respuesta 2: Sí, aunque lo pondría más visible.

Respuesta 3: Sí.

Respuesta 4: Sin ningún problema.

Respuesta 5: Sí, aunque fue difícil de encontrar.

Respuesta 6: Sin problema.

5. *Sobre la solicitud de una nueva reserva. ¿Le pareció sencillo el proceso de solicitud de una reserva? ¿Cómo lo mejoraría?*

Respuesta 1: Muy sencillo.

Respuesta 2: Sí, correcto.

Respuesta 3: Sí.

Respuesta 4: Sí.

Respuesta 5: Correcto.

Respuesta 6: Maravilloso.

6. *Sobre el panel de administración. ¿Encontró fácilmente todas las opciones del Gestor de Reservas?*

Respuesta 1: Sí.

Respuesta 2: Sí, todo correcto.

Respuesta 3: Sí, el menú muestra todas las operaciones que se pueden realizar con la aplicación de manera sencilla.

Respuesta 4: Sin ningún problema.

Respuesta 5: Sí.

Respuesta 6: Sí.

7. *Añada una nueva reserva al sistema. ¿Qué le ha parecido el proceso? ¿Encontró algún problema?*

Respuesta 1: Es sencillo, pero no entiendo bien qué fecha se asigna a las reservas cuatrimestrales.

Respuesta 2: Ok.

Respuesta 3: no encontré ningún problema.

Respuesta 4: Los pasos del proceso de reserva son intuitivos y guían al usuario.

Respuesta 5: El proceso es intuitivo, ha sido muy fácil añadir la reserva.

Respuesta 6: No entiendo a qué se refiere (en la elección del aula) cuando dice que tenga en cuenta el número de puestos de trabajo disponibles.

8. *Añada otro elemento al sistema: un aula, un grado, etcétera. ¿Es sencillo? ¿Cómo mejoraría este proceso?*

Respuesta 1: Muy sencillo.

Respuesta 2: Tuve problemas con las fechas al utilizar Firefox.

Respuesta 3: Es sencillo e intuitivo.

Respuesta 4: Es un proceso sencillo.

Respuesta 5: El proceso es sencillo.

Respuesta 6: Me parece suficientemente intuitivo.

9. *Modifique los valores de una reserva ya realizada e indique si le resultó fácil e intuitivo llevarlo a cabo.*

Respuesta 1: Sí, fácil.

Respuesta 2: Ok, correcto. Cambié el grupo y todo fue bien.

Respuesta 3: es fácil e intuitivo la edición de reservas realizadas.

Respuesta 4: Al modificar los datos el proceso es sencillo.

Respuesta 5: Es fácil e intuitivo.

Respuesta 6: Me parece muy intuitivo, aunque al editar los valores de la hora, quitaría los segundos.

10. *¿Le parece útil la información proporcionada en los informes y estadísticas?*

Respuesta 1: Sí.

Respuesta 2: Sí. Yo añadiría un cuadro con estadísticas.

Respuesta 3: Es muy útil la información ya que muestra un resumen por aula y asignatura.

Respuesta 4: Sí, quizá se podría añadir un desglose de horas/aula.

Respuesta 5: Es útil, aunque estaría bien añadir estadísticas sobre las horas de las aulas para que un profesor decida elegir una clase u otra a la hora de hacer la reserva.

Respuesta 6: Me parece útil, aunque incluiría también estadísticas por aula.

11. *Sobre el menú de opciones del gestor de reservas. ¿Le parece sencillo y lo suficientemente personalizable?*

Respuesta 1: Sí.

Respuesta 2: Correcto. Perfecto.

Respuesta 3: Es sencillo y se dispone de todas las opciones necesarias para configurar la herramienta.

Respuesta 4: Sí, se puede personalizar lo necesario.

Respuesta 5: Sí.

Respuesta 6: Me parece muy sencillo y personalizable.

12. *¿Qué le parece el proceso de confirmación de las reservas realizadas por los profesores? ¿Cómo lo mejoraría?*

Respuesta 1: Me gustaría poder seleccionar solo las reservas confirmadas durante la gestión.

Respuesta 2: Correcto. Aunque lo mejoraría pudiendo ver las reservas antes de confirmarlas.

Respuesta 3: Me parece correcto.

Respuesta 4: Me gusta.

Respuesta 5: Aceptable.

Respuesta 6: Me gustaría poder ver las reservas antes de confirmarlas.

13. *Tras realizar el test, ¿qué le parece la aplicación de Gestión de Reservas?*

Respuesta 1: Perfecta, porque ahorra muchísimos pasos y mucho tiempo con respecto a lo que se utilizaba antes.

Respuesta 2: Muy buena, va a mejorar mucho nuestro trabajo y a facilitar la comunicación de las reservas.

Respuesta 3: Es una herramienta útil y necesaria para el LDI.

Respuesta 4: Muy útil, además de sencilla de manejar.

Respuesta 5: Es una aplicación con un buen funcionamiento. Es útil y potente para realizar su función.

Respuesta 6: Una aplicación que moderniza la antigua haciéndola más intuitiva.

En definitiva, y tal como podemos apreciar en las respuestas proporcionadas por aquellos que realizaron el test, parece que la aplicación funciona sin problemas y **cumple con las expectativas** tanto de los clientes tanto en el aspecto de la estabilidad como en el desempeño de sus funciones. Además se realizaron algunas sugerencias que se evaluaron para su inclusión en el sistema e incluso como **posibles trabajos futuros**.

Determinamos pues que la aplicación está lista para su paso a producción y que ha terminado su proceso de desarrollo, alcanzando el hito de la versión 1.0.

6.3. Matrices de trazabilidad

Matriz de trazabilidad entre requisitos y pruebas unitarias

		PRUEBAS DE CAJA NEGRA												
		PU-N001	PU-N002	PU-N003	PU-N004	PU-N005	PU-N006	PU-N007	PU-N008	PU-N009	PU-N010	PU-N011	PU-N012	PU-N013
REQUISITOS	UR-C001		X										X	X
	UR-C002		X										X	X
	UR-C003		X										X	
	UR-C004			X	X								X	
	UR-C005			X	X								X	
	UR-C006	X											X	
	UR-C007			X		X							X	
	UR-C008			X			X						X	
	UR-C009			X				X					X	
	UR-C010			X					X				X	
	UR-C011			X						X			X	
	UR-C012			X							X		X	X
	UR-C013			X							X		X	X
	UR-C014			X							X		X	X
	UR-C015													
	UI-C016													
	UI-C017	X											X	
	UI-C018	X											X	
	UI-C019			X	X								X	
	UI-C020			X		X							X	
	UI-C021			X			X						X	
	UI-C022			X				X					X	
	UI-C023			X					X				X	
	UI-C024			X						X			X	
	UI-C025			X								X	X	
	UK-R001	X	X		X	X	X	X	X	X	X	X		
	UK-R002	X	X		X	X	X	X	X	X	X	X		
	UR-R003													
US-R004														
UR-R005														
UI-R006	X	X		X	X	X	X	X	X	X	X			
UI-R007													X	
UI-R008										X				

Tabla 163: Matriz de trazabilidad 1

		PRUEBAS DE CAJA NEGRA												
		PU-N014	PU-N015	PU-N016	PU-N017	PU-N018	PU-N019	PU-N020	PU-N021	PU-N022	PU-N023	PU-N024	PU-N025	PU-N026
REQUISITOS	UR-C001	X		X	X		X		X		X		X	
	UR-C002	X		X	X		X		X		X		X	
	UR-C003	X	X	X	X		X		X		X		X	
	UR-C004	X		X	X		X		X		X		X	
	UR-C005	X	X	X	X		X		X		X		X	
	UR-C006	X			X		X		X		X		X	
	UR-C007				X	X								
	UR-C008						X	X						
	UR-C009								X	X				
	UR-C010										X	X		
	UR-C011												X	X
	UR-C012													
	UR-C013													
	UR-C014													
	UR-C015													
	UI-C016													
	UI-C017	X			X		X		X					
	UI-C018	X			X		X		X					
	UI-C019	X			X		X		X					
	UI-C020				X									
	UI-C021						X							
	UI-C022								X					
	UI-C023										X			
	UI-C024												X	
	UI-C025													
	UK-R001													
	UK-R002													
	UR-R003													
	US-R004													
	UR-R005													
	UI-R006													
	UI-R007	X	X		X		X		X		X		X	
UI-R008														

Tabla 164: Matriz de trazabilidad 2

PRUEBAS DE CAJA NEGRA													
R E Q U I S I T O S		PU- N027	PU- N028	PU- N029									
	UR-C001	X											
	UR-C002	X											
	UR-C003	X	X										
	UR-C004	X	X										
	UR-C005		X										
	UR-C006	X											
	UR-C007												
	UR-C008												
	UR-C009												
	UR-C010												
	UR-C011												
	UR-C012												
	UR-C013												
	UR-C014												
	UR-C015												
	UI-C016												
	UI-C017	X											
	UI-C018	X											
	UI-C019												
	UI-C020												
	UI-C021												
	UI-C022												
	UI-C023												
	UI-C024												
	UI-C025			X									
	UK-R001												
	UK-R002												
	UR-R003												
	US-R004												
	UR-R005												
	UI-R006												
	UI-R007		X	X									
	UI-R008												

Tabla 165: Matriz de trazabilidad 3

Matriz de trazabilidad entre requisitos y pruebas de integración

		PRUEBAS DE INTEGRACIÓN												
		PI-001	PI-002	PI-003	PI-004	PI-005	PI-006	PI-007	PI-008	PI-009	PI-010	PI-011	PI-012	PI-013
R E Q U I S I T O S	UR-C001			X	X									
	UR-C002			X	X									
	UR-C003			X	X									
	UR-C004					X	X	X						
	UR-C005					X	X							
	UR-C006	X	X											
	UR-C007								X	X				
	UR-C008										X	X		
	UR-C009												X	X
	UR-C010													
	UR-C011													
	UR-C012													
	UR-C013													
	UR-C014													
	UR-C015													
	UI-C016													
	UI-C017	X												
	UI-C018		X											
	UI-C019					X	X	X						
	UI-C020								X	X				
	UI-C021										X	X		
	UI-C022												X	X
	UI-C023													
	UI-C024													
	UI-C025													
	UK-R001	X	X	X										
	UK-R002	X	X	X										
	UR-R003													
	US-R004													
	UR-R005													
	UI-R006	X	X	X										
	UI-R007													
	UI-R008													

Tabla 166: Matriz de trazabilidad 4

		PRUEBAS DE INTEGRACIÓN											
		PI-014	PI-015	PI-016	PI-017	PI-018							
REQUISITOS	UR-C001												
	UR-C002												
	UR-C003												
	UR-C004												
	UR-C005												
	UR-C006												
	UR-C007												
	UR-C008												
	UR-C009												
	UR-C010	X	X										
	UR-C011			X									
	UR-C012					X							
	UR-C013					X							
	UR-C014					X							
	UR-C015												
	UI-C016												
	UI-C017												
	UI-C018												
	UI-C019												
	UI-C020												
	UI-C021												
	UI-C022												
	UI-C023	X	X										
	UI-C024			X									
	UI-C025				X								
	UK-R001												
	UK-R002												
	UR-R003												
	US-R004												
	UR-R005												
	UI-R006												
	UI-R007												
	UI-R008												

Tabla 167: Matriz de trazabilidad 5

Matriz de trazabilidad entre requisitos y pruebas de implantación

		PRUEBAS DE IMPLANTACIÓN											
		PIS-001	PIS-002	PIS-003									
R E Q U I S I T O S	UR-C001	X	X	X									
	UR-C002	X	X	X									
	UR-C003	X	X	X									
	UR-C004	X	X										
	UR-C005	X	X										
	UR-C006	X	X										
	UR-C007	X	X										
	UR-C008	X	X										
	UR-C009	X	X										
	UR-C010	X	X										
	UR-C011	X	X										
	UR-C012	X	X										
	UR-C013	X	X										
	UR-C014	X	X	X									
	UR-C015	X	X	X									
	UI-C016	X											
	UI-C017	X	X	X									
	UI-C018	X	X	X									
	UI-C019	X	X										
	UI-C020	X	X										
	UI-C021	X	X										
	UI-C022	X	X										
	UI-C023	X	X										
	UI-C024	X	X										
	UI-C025	X	X										
	UK-R001	X	X										
	UK-R002	X	X										
	UR-R003	X	X	X									
	US-R004	X	X	X									
	UR-R005	X	X	X									
	UI-R006	X	X										
	UI-R007	X	X										
	UI-R008	X	X										

Tabla 168: Matriz de trazabilidad 6

7. Planificación y presupuesto

7.1. Planificación

La **planificación de este proyecto** viene firmemente marcada por los requerimientos del Trabajo de Fin de Grado, que tiene un peso de 12ETS, cada uno de los cuales equivale a entre 25 y 30 horas de trabajo. Teniendo este requisito en cuenta y proponiendo una organización de no más de 20 horas de trabajo a la semana tendríamos un proyecto que se extendería en el tiempo unas 18 semanas.

Además de esto y para garantizar que el proyecto termina en el plazo para su presentación en la segunda convocatoria de TFG de la Universidad (cuya solicitud comienza el 5 de mayo) y con el fin también de evitar el parón por el periodo de exámenes de mayo, se decide proponer como fecha de finalización del proyecto la primera semana de marzo. Por otro lado, también se tendrá en cuenta en la planificación el parón por las fiestas navideñas y la realización de prácticas y exámenes por las asignaturas del primer cuatrimestre.

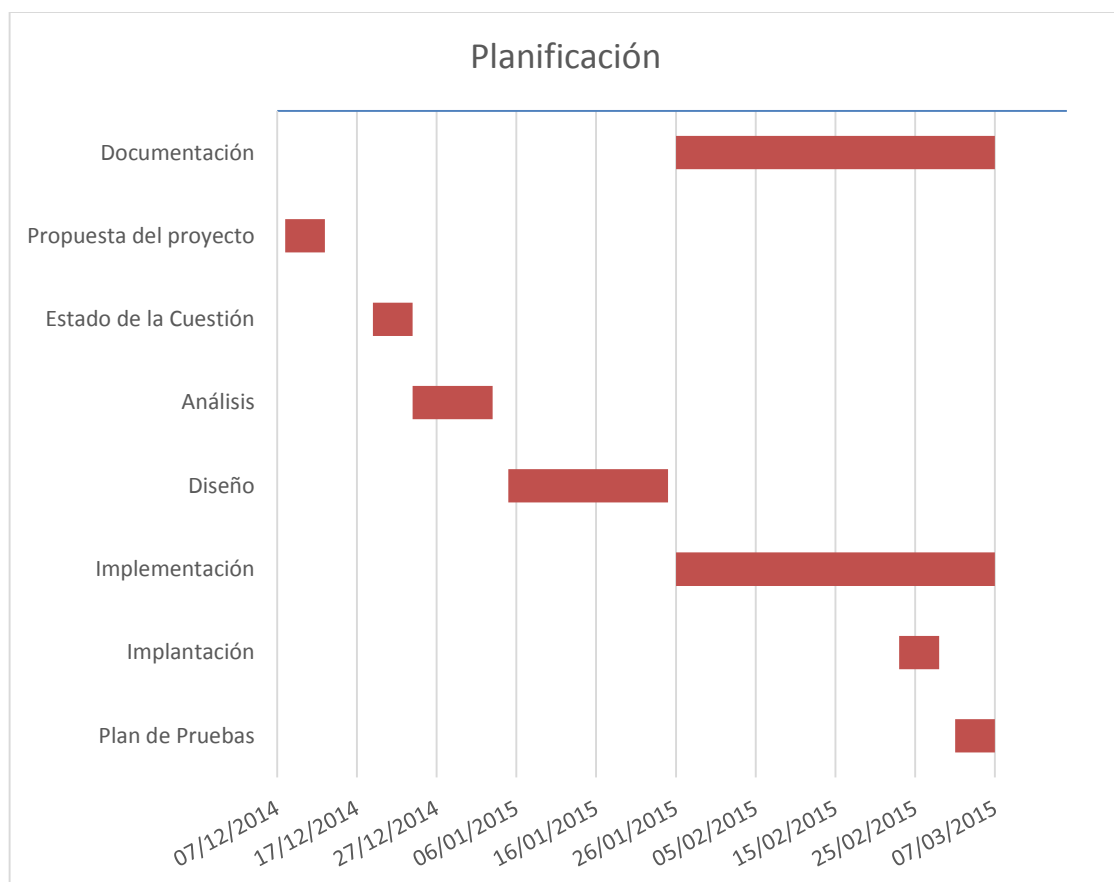


Figura 35: Planificación

Con todos estos datos sobre la mesa, **se propone la siguiente planificación del proyecto**, comenzando el día 8 de diciembre de 2014 y con fecha esperada de finalización el 9 de marzo de 2015:

7.2. Presupuesto

A partir de la planificación del apartado anterior podemos realizar una **estimación del coste total del proyecto** teniendo en cuenta los siguientes factores:

- **Salarios.** Para la realización de este proyecto será necesario un equipo de al menos tres personas que cubran los roles de Administrador de Sistemas, Analista diseñador y Programador.
- **Amortizaciones.** Será necesario tener en cuenta todo el material necesario para la realización del proyecto y su influencia en el coste total del proyecto.
- **Costes.** Teniendo en cuenta los salarios y el gasto en amortizaciones de material podemos hacer un cálculo aproximado del coste final del proyecto.
- **Precio final del proyecto.** Calcularemos el precio final teniendo en cuenta los costes, así como el margen de seguridad y la inclusión de un margen de beneficio.

A continuación se detalla cada uno de estos elementos de forma más precisa.

Salarios

A continuación se muestra una tabla que desglosa los costes empresariales asociados al salario de los empleados, teniendo en cuenta impuestos y retenciones a la seguridad social:

Cálculo de Costes Empresariales			
Datos del Empleado			
Puesto	Administrador de Sistemas	Analista de sistemas	Programador
Jornada Laboral	4	4	4
Horas Semanales	20	20	20
Retribuciones Brutas			
Retribución Anual	12.306,00 €	14.562,00 €	10.284,00 €
Salario Base/Mes	1.025,50 €	1.213,50 €	857,00 €
Pluses			
Salario Neto/Mes	1.025,50 €	1.213,50 €	857,00 €
Salario Neto/Año	14.357,00 €	16.989,00 €	11.998,00 €
Salario Cotizable	1.025,50 €	1.213,50 €	857,00 €
Pagas Extra	170,92 €	202,25 €	142,83 €
Base de Cotización	1.196,42 €	1.415,75 €	999,83 €
Impuesto sobre la Renta de las Personas Físicas			
Retención mensual IRPF	246,12 €	291,24 €	205,68 €
Retención anual IRPF	2.953,44 €	3.494,88 €	2.468,16 €
Retenciones de la Seguridad Social			
Contingencias Comunes (9,8%)	117,25 €	138,74 €	97,98 €
Accidentes Profesionales (1,8%)	21,54 €	25,48 €	18,00 €
Desempleo (6%)	71,79 €	84,95 €	59,99 €
Formación Profesional (0,6%)	7,18 €	8,49 €	6,00 €
Aportación FOGASA (0,4%)	4,79 €	5,66 €	4,00 €
Total Seguridad Social/Mes	222,53 €	263,33 €	185,97 €
Total Seguridad Social/Año	2.670,40 €	3.159,95 €	2.231,63 €
Salario Bruto Anual	19.980,84 €	23.643,83 €	16.697,79 €
Coste de Hora Trabajada	20,81 €	24,63 €	17,39 €

Tabla 169: Cálculo de costes empresariales

Amortizaciones

Se incluyen los equipos necesarios para la realización del proyecto y su correspondiente gasto en amortización para el proyecto, así como diverso material fungible necesario, pese a que su uso e influencia en el precio final es casi anecdótico:

Costes de Equipos				
Concepto	Modelo	Cantidad	Precio Ud.	Total
Ordenador Sobremesa	Acer Aspire TC-605	1	599,00 €	599,00 €
Monitor	BenQ GL2023A 19.5" LED	3	77,95 €	233,85 €
Impresora	Xerox WorkCentre™ 6015	1	190,00 €	190,00 €
Total				1.022,85 €

Tabla 170: Costes de equipos

Amortizaciones		
Tiempo (Semanas)	Amortización Semanal	Base Imponible
18	33,28 €	465,89 €
18	12,99 €	181,88 €
18	10,56 €	147,78 €
Totales	56,83 €	795,55 €

Tabla 171: Amortizaciones

Material Fungible				
Actividad	Concepto	Cantidad	Precio Ud.	Total
A1.- Documentación	Paquete Folios 500	4	2,50 €	10,00 €
A2.- Propuesta del proyecto				0,00 €
A3.- Estado de la Cuestión				0,00 €
A4.- Análisis				0,00 €
A5.- Diseño	Post-It	14	0,80 €	11,20 €
A6.- Implementación				0,00 €
A7.- Implantación				0,00 €
A8.- Plan de Pruebas				0,00 €
Total				21,20 €

Tabla 172: Material fungible

Costes

Teniendo en cuenta lo anterior se calcula el coste real del proyecto incluyendo gastos tanto en material como en personal.

Gastos de Personal Imputables al Proyecto			
Datos del Empleado			
Puesto	Administrador de Sistemas	Analista de sistemas	Programador
Horas Semanales	20	20	20
Desglose por Categorías y Actividades (Participación en semanas de cada empleado y coste para la empresa)			
A1.- Documentación	4,00	4,00	0,00
A2.- Propuesta del proyecto	0,00	1,00	0,00
A3.- Estado de la Cuestión	0,00	1,00	0,00
A4.- Análisis	2,00	2,00	0,00
A5.- Diseño	4,00	4,00	0,00
A6.- Implementación	0,00	0,00	8,00
A7.- Implantación	1,00	1,00	1,00
A8.- Plan de Pruebas	1,00	1,00	1,00
Total Semanas Invertidas	12,00	14,00	10,00
Coste Hora Trabajada	20,81 €	24,63 €	17,39 €
Total Coste	4.995,21 €	6.896,12 €	3.478,71 €

Presupuesto del Proyecto (Costes de la Empresa)			
Desglose por Actividades			
	Personal	Equipos	Fungible
A1.- Documentación	3.635,39 €	113,65 €	10,00 €
A2.- Propuesta del proyecto	492,58 €	113,65 €	0,00 €
A3.- Estado de la Cuestión	492,58 €	113,65 €	0,00 €
A4.- Análisis	1.817,69 €	113,65 €	11,20 €
A5.- Diseño	3.635,39 €	113,65 €	0,00 €
A6.- Implementación	2.782,96 €	113,65 €	0,00 €
A7.- Implantación	1.256,72 €	113,65 €	0,00 €
A8.- Plan de Pruebas	1.256,72 €	0,00 €	0,00 €
Total	15.370,03 €	795,55 €	21,20 €

Tabla 173: Gastos y Presupuesto

Análisis, diseño e implantación de una aplicación para la gestión de reservas para las aulas docentes del departamento de informática

Precio final del proyecto

El precio final del proyecto lo determinará el coste además del margen de seguridad y un margen adicional de beneficio, y será de **VEINTIDÓS MIL CIENTO CATORCE EUROS CON TREINTA Y NUEVE CÉNTIMOS**.

Precio del Proyecto
Costes Empresariales
16.186,78 €
Costes Indirectos (8%)
1.294,94 €
Margen de Seguridad (10%)
1.748,17 €
Margen de Beneficio (15%)
2.884,49 €
Precio Final
22.114,39 €

Tabla 174: Precio del Proyecto

7.3. Esfuerzo Estimado vs Esfuerzo Real

Una vez terminado el proyecto podemos realizar una comparación entre la planificación y los costes previstos al inicio del desarrollo de este proyecto y el **esfuerzo real** que se ha realizado finalmente.

En lo relacionado con la **planificación del sistema**, la necesidad del aprendizaje de nuevas tecnologías como PHP o conocimientos sobre Linux, la virtualización y la administración de servidores, provocaron algunas alteraciones en la planificación, la cual se expuso al principio del documento y que se corresponde con el siguiente gráfico:

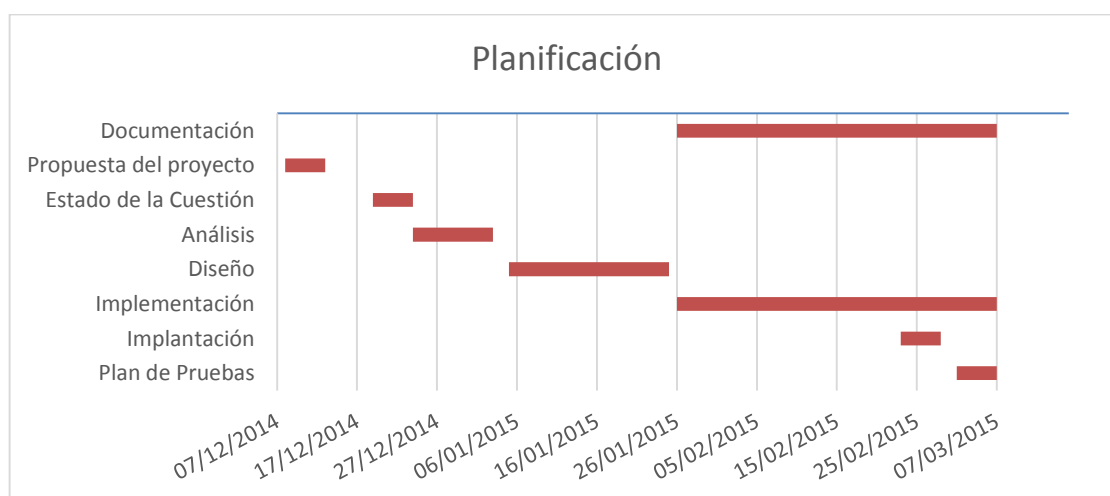


Figura 36: Planificación estimada

Análisis, diseño e implantación de una aplicación para la gestión de reservas para las aulas docentes del departamento de informática

Estas alteraciones provocaron un incumplimiento de dicha planificación alterando las fechas de finalización de algunas de las tareas realizadas, como la *Implementación* (que afectó a su vez a la *Implantación* y el *Plan de Pruebas*) y la redacción de la *Documentación* asociada a dicho proceso. En definitiva, la planificación real resultó ser la siguiente:



Figura 37: Planificación real

En total se sufrió un **retraso de 4 semanas**, tiempo considerablemente elevado y que pudo provocar bastantes problemas de no ser porque en las fechas de inicio y fin, la planificación **contempló la posibilidad de retrasos** que pudiesen provocar exámenes, vacaciones (debido a que el cierre de la Universidad) y otros imprevistos, estableciendo una fecha de finalización muy anterior a la entrega y presentación definitiva del TFG. Gracias a esto, el escollo de los retrasos puede **considerarse un mal menor**.

Donde el retraso si supone un **problema a tener en cuenta** es en el número de horas trabajadas y, por ende, en el coste real del proyecto. Esto provoca los siguientes cambios en el cálculo de costes expuesto anteriormente en el documento:

Gastos de Personal Imputables al Proyecto			
Datos del Empleado			
Puesto	Administrador de Sistemas	Analista de sistemas	Programador
Horas Semanales	20	20	20
Desglose por Categorías y Actividades (Participación en semanas de cada empleado y coste para la empresa)			
A1.- Documentación	6,00	6,00	0,00
A2.- Propuesta del proyecto	0,00	1,00	0,00
A3.- Estado de la Cuestión	0,00	1,00	0,00
A4.- Análisis	2,00	2,00	0,00
A5.- Diseño	4,00	4,00	0,00
A6.- Implementación	0,00	0,00	12,00
A7.- Implantación	1,00	1,00	1,00
A8.- Plan de Pruebas	1,00	1,00	1,00
Total Semanas Invertidas	14,00	16,00	14,00
Coste Hora Trabajada	20,81 €	24,63 €	17,39 €
Total Coste	5.827,75 €	7.881,28 €	4.870,19 €

Presupuesto del Proyecto (Costes de la Empresa)			
Desglose por Actividades			
	Personal	Equipos	Fungible
A1.- Documentación	5.453,08 €	113,65 €	10,00 €
A2.- Propuesta del proyecto	492,58 €	113,65 €	0,00 €
A3.- Estado de la Cuestión	492,58 €	113,65 €	0,00 €
A4.- Análisis	1.817,69 €	113,65 €	11,20 €
A5.- Diseño	3.635,39 €	113,65 €	0,00 €
A6.- Implementación	4.174,45 €	113,65 €	0,00 €
A7.- Implantación	1.256,72 €	113,65 €	0,00 €
A8.- Plan de Pruebas	1.256,72 €	0,00 €	0,00 €
Total	18.579,21 €	795,55 €	21,20 €

Figura 38: Coste de personal

De esta forma, el coste final del proyecto ajustado a la planificación y a los costes de personal reales es de **VEINTE MIL NOVECIENTOS CUARENTA Y SIETE EUROS CON SESENTA Y CUATRO CÉNTIMOS** (20.947,64€):

Precio del Proyecto
Costes Empresariales
19.395,96 €
Costes Indirectos (8%)
1.551,68 €
Precio Final
20.947,64 €

Tabla 175: Precio real

Este retraso supone pues, un **incremento en el coste del proyecto** de 3.465,91€, superior a nuestro margen de seguridad del 10% y que por lo tanto provoca una reducción de nuestros beneficios previstos.

Precio del Proyecto A priori	Precio del Proyecto Ajustado
Costes Empresariales	Costes Empresariales
16.186,78 €	19.395,96 €
Costes Indirectos (8%)	Costes indirectos (8%)
1.294,94 €	1.551,68 €
Margen de Seguridad (10%)	Costes Empresariales
1.748,17 €	-
Margen de Beneficio (15%)	Costes Indirectos (8%)
2.884,49 €	-
Precio Final	Precio Final
22.114,39 €	20.947,64 €

Tabla 176: Precio real vs. estimado

En definitiva, la realización de este proyecto supondría un **beneficio real de 1.166,75€**, un 40% del beneficio esperado.

Esto evidencia la importancia de una buena planificación en un proyecto. En este caso la realización de **una planificación aceptable** (errónea, no obstante) nos ha permitido obtener beneficios, pero un ligero fallo en ella ha provocado un **aumento de los costes considerable**. La falta de experiencia en la planificación de proyectos es uno de los motivos de la diferencia entre lo planificado y lo real, aunque supone una valiosa experiencia aprendida de cara a proyectos futuros.

8. Conclusiones y trabajos futuros

8.1. Conclusiones

Producto

El Trabajo de Fin de Grado recogido en este documento (*Análisis, diseño e implantación de una aplicación para la gestión de reservas para las aulas docentes del departamento de informática*) ha consistido en el diseño técnico e implementación de una aplicación web para la gestión de las aulas docentes del Laboratorio del Departamento de Informática de la Universidad Carlos III de Madrid. Dicho software se ha desarrollado como un plugin de WordPress capaz de funcionar con cualquier versión de este CMS superior a la 4.0.

Al comienzo de este documento establecimos una serie de objetivos que la aplicación debía cumplir para satisfacer los requisitos del cliente. Ahora podemos decir que estos objetivos han sido cumplidos porque:

- **El proceso de gestión de las reservas** es ahora más fácil con respecto a la aplicación original, y nuestra nueva web es compatible con dispositivos móviles.
- **La gestión no se limita solo a reservas y aulas.** El sistema puede gestionar información sobre grados, asignaturas, software y máquinas virtuales para ofrecer a los técnicos una herramienta más sencilla para controlar todos los elementos relacionados con las aulas.
- **Los profesores pueden solicitar reservas** con un formulario protegido mediante el servidor LDAP de la universidad.
- **Los técnicos pueden ver en cada momento estadísticas** sobre las aulas y pueden descargarlas en un documento de Microsoft Excel.

Además, al comienzo del desarrollo, la última versión de WordPress era la 4.0. En el momento de escribir este documento, la versión más actualizada es la 4.2, y el proceso de actualización no requirió ningún cambio en el sistema desarrollado. Esto demuestra que la aplicación es compatible y **actualizable de forma sencilla y cómoda**.

Proceso

El desarrollo del proyecto ha sido, en líneas generales, un proceso bastante satisfactorio y que se ha realizado sin demasiados incidentes.

El incidente más relevante es el retraso de 4 semanas sufrido con respecto a la planificación original, tal y como se explica en el apartado 7.3. *Esfuerzo Estimado vs Esfuerzo Real*. Dicho retraso se debió principalmente a una mala estimación del tiempo necesario para el aprendizaje de algunas de las nuevas tecnologías que era necesario aplicar en el proyecto, y pese a que no causó verdaderos problemas en cuanto a fechas de entrega y presentación del proyecto (ya que la planificación se realizó asegurando un buen margen de tiempo previendo dichos retrasos) sí causó un problema de presupuesto.

El **presupuesto real** supuso un incremento de 3.465,91€, superior al margen de seguridad del 10% establecido y que por lo tanto provocó una reducción de los beneficios económicos del proyecto.

Este retraso evidencia la gran importancia que una correcta planificación tiene sobre un proyecto. Si bien podemos calificar nuestra planificación de aceptable (pues se cumplió el objetivo final, entregar el proyecto a tiempo, y además se obtuvieron beneficios), lo cierto es que ni la fecha de entrega prevista ni el margen de seguridad establecido fueron suficientemente realistas. Esto podría ser provocado por una falta de experiencia, y en cualquier caso supone una valiosa lección de cara a la realización de proyectos en el futuro.

Personales

En lo personal, estoy muy orgulloso de este proyecto. Éste no solo demuestra mis capacidades para convertirme en un ingeniero, sino también muestra que soy capaz de planificar un proyecto profesional y aprender nuevas tecnologías por mí mismo.

Para la realización del trabajo fueron muy importantes los **contenidos aprendidos en asignaturas** como Interfaces de Usuario, Tecnologías Informáticas para la Web para el desarrollo de la página web; de Ingeniería de la Seguridad y Principios de Desarrollo del Software para la fase de pruebas; de Sistemas Operativos, Sistemas Distribuidos, Diseño de Sistemas Operativos y Ficheros y Bases de datos para la configuración y establecimiento del

servidor; y de Ingeniería del Software y Dirección de Proyectos de Desarrollo del Software para generar toda la documentación del proyecto.

También fue necesario **aprender nuevas tecnologías** que no se enseñan durante la carrera y profundizar en otras para tener un conocimiento más específico. Así, durante el desarrollo del proyecto aprendí tecnologías como PHP o conocimientos sobre Linux, la virtualización y la administración de servidores

Me parece especialmente relevante que el sistema desarrollado vaya a utilizarse dentro del ámbito de la Universidad de forma práctica, ya que para mí es un reconocimiento de la calidad del mismo. Además, me alegra poder facilitar la labor de los técnicos del Laboratorio del Departamento de Informática, ya que durante mi periodo de trabajo en él y la realización de este proyecto ellos siempre han mostrado su voluntad de ayudarme a mí en mi formación. Poder devolver de alguna forma el favor con mi proyecto supone una satisfacción extra, que se une a la de haber logrado todos los objetivos de funcionalidad y calidad fijados para este proyecto.

Gestor de Reservas es una aplicación genial y estoy seguro de que será usada durante mucho tiempo dentro del Laboratorio del Departamento de Informática.

8.2. Trabajos futuros

Por último, se proponen algunas ideas sobre posibles nuevas características que sería interesante ver implementadas en la aplicación pero que por diversos motivos no pudieron verse en la versión final:

1. **Accesibilidad.** Asegurarse de que el proyecto cumple con todas las recomendaciones para hacer del sitio web un entorno accesible a personas con algún tipo de discapacidad.
2. **Mayor capacidad de configuración para un uso comercial.** La aplicación ha sido diseñada para el Laboratorio del Departamento de Informática de la universidad, pero sería posible aplicar algunos cambios como la capacidad de personalizar las tablas de la base de datos, para ofrecer el proyecto como una solución comercial que cualquier organización pudiera usar para gestionar sus aulas (como gimnasios, escuelas o quizás otros departamentos de la universidad).
3. **Incluir gestión de incidencias y equipo informático.** La aplicación desarrollada es un completo centro de gestión de las aulas y sería

posible mejorarla incluyendo la posibilidad de usarla para registrar incidencias y problemas en los equipos además de mantener un registro sobre el equipo disponible en cada clase como auriculares, teclados o ratones.

4. **Un sistema de estadísticas más completo.** El sistema desarrollado incluye un generador de estadísticas con todos los datos que los técnicos de laboratorio deben enviar al departamento sobre la ocupación de las aulas. Sin embargo sería una buena idea implementar mejores estadísticas para uso interno con información sobre las clases favoritas de los profesores, las asignaturas más frecuentes y más.
5. **Realización de una versión en forma de aplicación para móviles.** El sistema desarrollado tiene una interfaz web adaptable a smartphone y tablets, pero de por si el hecho de que sea una web puede provocar algunas incomodidades durante el trabajo que desarrollan los técnicos. Esto podría solventarse desarrollando una aplicación específica para Android o iOS que se centre solo en las tareas de gestión de aulas.
6. **Paneles de información para estudiantes.** Debido a que toda la información sobre las aulas docentes se encuentra centralizada ahora en la aplicación, podría ser interesante la inclusión de paneles en las aulas que vayan mostrando: la ocupación de las aulas, el software disponible, próximas clases, etcétera; por medio de un monitor conectado a algún tipo de placa que reciba la información tipo Arduino o Raspberry Pi.
7. **Gestión de los equipos informáticos.** Dado que se pretende que esta aplicación centralice todas las tareas de gestión de las aulas, otra propuesta de trabajo futuro es integrar otra de las herramientas desarrolladas por el laboratorio para este fin. Esta herramienta consiste en una serie de scripts para encender/apagar los equipos de las aulas de forma remota. Estos scripts podrían portarse a una interfaz web dentro de esta aplicación.

Apéndice I: Manual de administración

El Gestor de Reservas es un plugin para WordPress desarrollado en PHP y HTML que permite la gestión de las aulas docentes del Laboratorio del Departamento de Informática de la Universidad Carlos III de Madrid, de forma sencilla y automatizada. Este manual le servirá como guía para el aprendizaje y utilización del Gestor de Reservas y está destinado a los administradores del sistema.

Instalar la aplicación

Como cualquier plugin de Wordpress, la instalación de la aplicación se realiza descomprimiendo el contenido de la aplicación en la carpeta "plugin" de Wordpress y posteriormente activándolo en la sección correspondiente en el panel de administración de Wordpress.

Una vez activado se mostrará un menú adicional en el menú de administración de Wordpress para la gestión de las reservas:

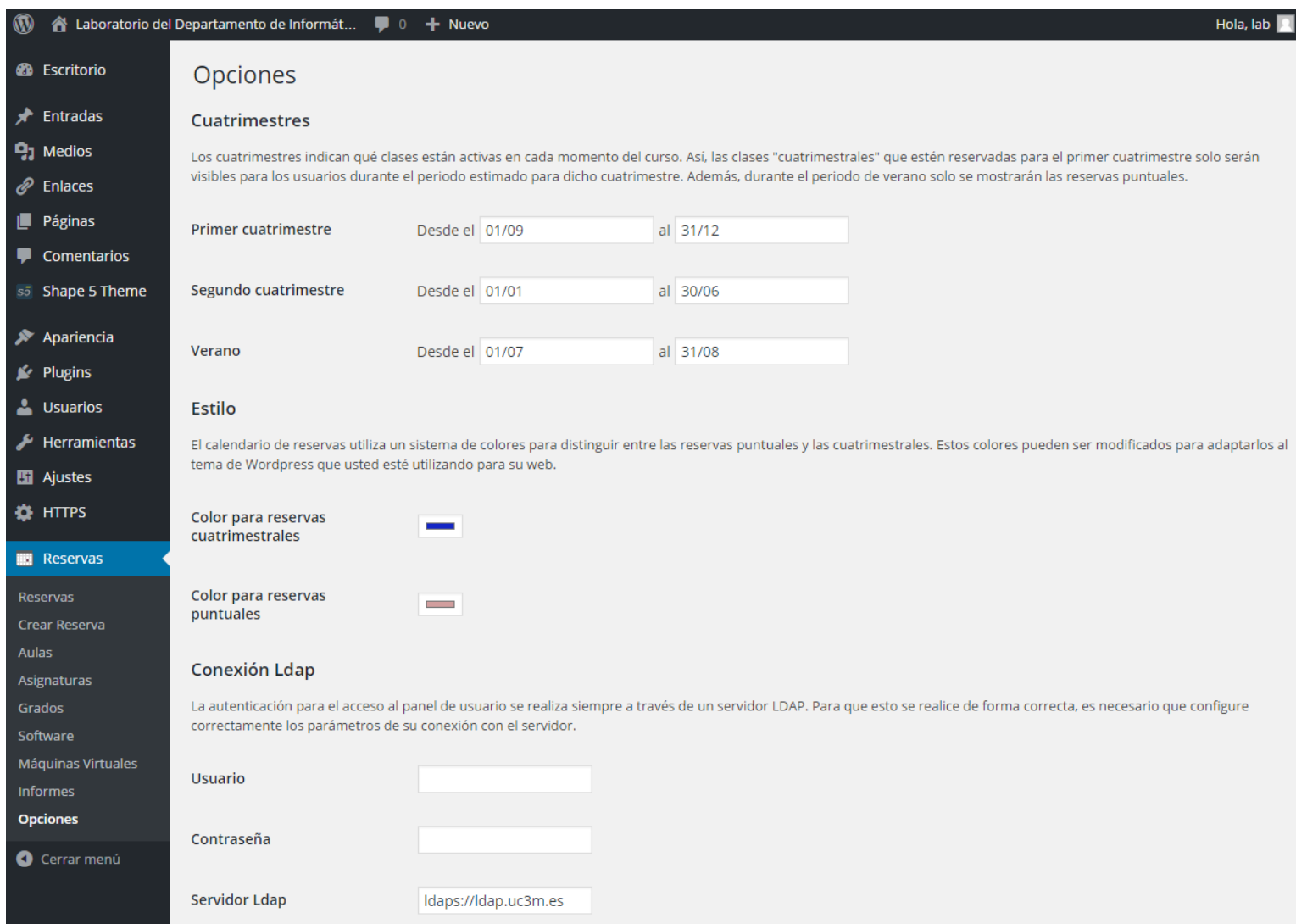


Configuración

La primera vez que se activa el plugin, el sistema cargará las opciones de configuración predeterminadas. Estos parámetros son editables desde la sección de Opciones del menú de Reservas, donde se pueden modificar los siguientes valores:

- Fechas de inicio y fin de cuatrimestre. Es posible cambiar el momento en el que empieza y acaba un cuatrimestre del curso académico, así como el periodo de verano.
- Estilo. Es posible cambiar los colores de las reservas cuatrimestrales y puntuales, eligiendo el color que mejor se adapte a la interfaz de la página web.
- Conexión LDAP. Para la solicitud de reservas es necesario autenticar con un servidor LDAP que debe ser configurado desde este lugar. Es necesario especificar el nombre de usuario, contraseña y dirección del servidor.

Adicionalmente, se incluye una pequeña ventana de ayuda con información sobre los *shortcodes*, de los que se hablará de forma más detallada en este manual más adelante.



Laboratorio del Departamento de Informát... 0 + Nuevo Hola, lab

Opciones



Cuatrimestres

Los cuatrimestres indican qué clases están activas en cada momento del curso. Así, las clases "cuatrimestrales" que estén reservadas para el primer cuatrimestre solo serán visibles para los usuarios durante el periodo estimado para dicho cuatrimestre. Además, durante el periodo de verano solo se mostrarán las reservas puntuales.

Primer cuatrimestre	Desde el	01/09	al	31/12
Segundo cuatrimestre	Desde el	01/01	al	30/06
Verano	Desde el	01/07	al	31/08

Estilo

El calendario de reservas utiliza un sistema de colores para distinguir entre las reservas puntuales y las cuatrimestrales. Estos colores pueden ser modificados para adaptarlos al tema de Wordpress que usted esté utilizando para su web.

Color para reservas cuatrimestrales	
Color para reservas puntuales	

Conexión Ldap

La autenticación para el acceso al panel de usuario se realiza siempre a través de un servidor LDAP. Para que esto se realice de forma correcta, es necesario que configure correctamente los parámetros de su conexión con el servidor.

Usuario	<input type="text"/>
Contraseña	<input type="password"/>
Servidor Ldap	<input type="text" value="ldaps://ldap.uc3m.es"/>

Gestión de reservas

La gestión de reservas se realiza a través de los menús de “Crear Reserva” y “Reservas”. Desde el primero pueden añadirse nuevas reservas al sistema mientras que en el segundo pueden verse y modificarse los valores de las reservas previamente añadidas al sistema.

Además, antes de realizar una reserva será necesario añadir información relativa a las aulas, los grados, asignaturas y demás información relacionada de la que va a depender directamente la reserva que se va a realizar.

El proceso de creación de reservas está dividido en tres pasos:

Crear una nueva reserva: Paso 1 de 3

Profesor

prueba

Correo electrónico del profesor que realiza la reserva.

Asignatura

Asignatura 3

Asignatura que se impartirá durante la sesión de la reserva.

Tipo de reserva

Cuatrimestral

Indica la periodicidad de la reserva: la puntual solo se hará efectiva para un único día, mientras que la cuatrimestral reserva el aula para todo el cuatrimestre.

Fecha

01/01/2015

Indique la fecha en que se desea realizar la reserva.

Siguiente

Selección de aula y horario: Paso 2 de 3

A continuación se muestra la ocupación de las aulas los Miércoles:

Hora	AulaGlobal	Aula 1	Aula 2	Aula 3	Aula 4	Aula 5
09:00						
09:15						
09:30						
09:45		Asignatura 3 (G11)			Asignatura 2 (G06)	
10:00		Grado 3 el 2015-09-01			Grado 2	
10:15						
10:30						
10:45						
11:00						
11:15						
11:30						
11:45						
12:00						
12:15						
12:30						
12:45						

A la izquierda puede verse la ocupación de las aulas del día en el que se desea programar la reserva. Por favor, compruebe que el horario elegido se corresponde con alguno de los huecos libres entre las reservas ya establecidas y complete los siguientes campos:

Grupo

11

Grupo de la asignatura que va a aprovechar dicha reserva.

Aula

AulaGlobal

Aula que se desea reservar. Tenga en cuenta el número de puestos de trabajo disponibles en cada aula.

Hora

09:00 - 11:00

Hora en la que se desea realizar la reserva.

Atrás

Siguiente

Crear una nueva reserva: Paso 3 de 3

Se añadirá al sistema una nueva reserva con los siguientes datos facilitados por el usuario. Por favor, confirme que los datos de la reserva son correctos antes de finalizar la solicitud:

Profesor: prueba

Asignatura: Asignatura 3

Grupo: 11

Tipo de Reserva: cuatrimestral

Día: Todos los Miércoles del 2 cuatrimestre

Aula: AulaGlobal

Horario: de 09:00 a 11:00

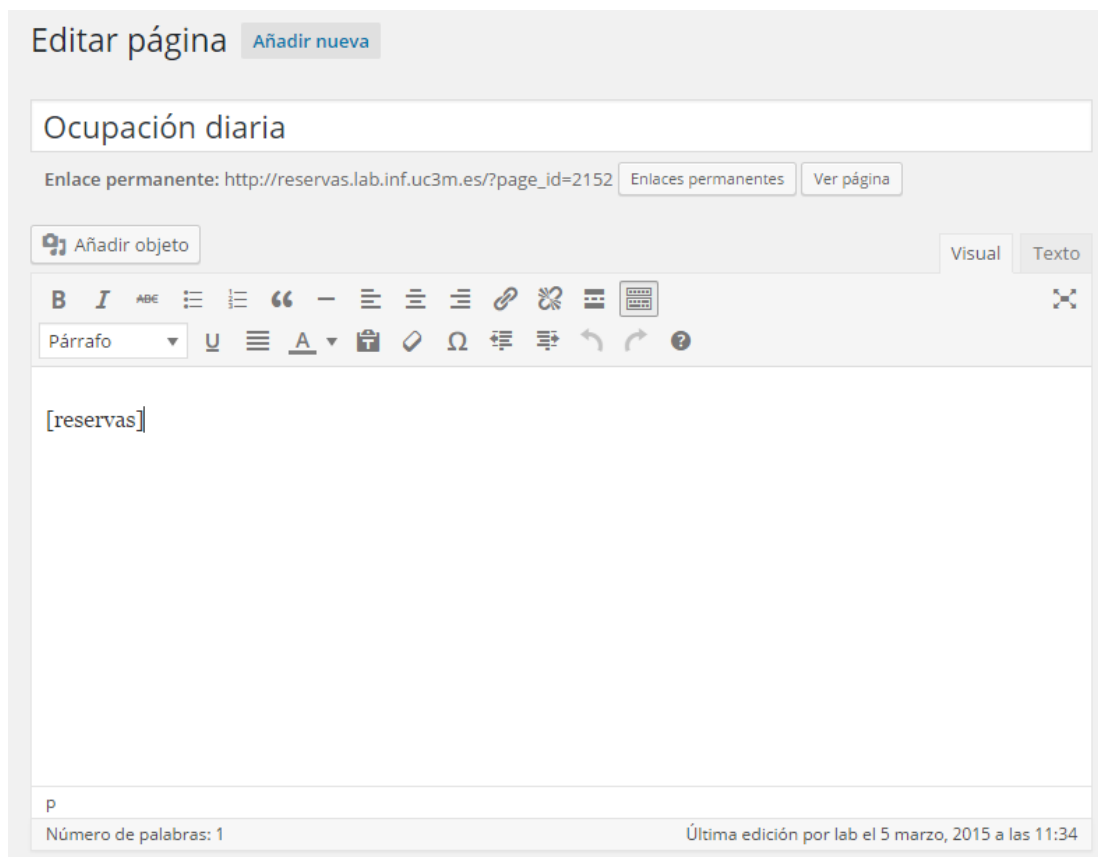
☐ Confirmar esta reserva. Solo las reservas confirmadas aparecen en el calendario de reservas visible para profesores y alumnos.

[Atrás](#)[Finalizar reserva](#)

Shortcodes

Los *shortcodes* son etiquetas usadas por WordPress para llamar a funciones desde las páginas del sitio. Esto permite que el contenido de una página se genere de forma dinámica. En el Gestor de Reservas los *shortcodes* se utilizan para invocar las diferentes vistas de las reservas (por día o por aula) así como el panel de solicitud de reservas para profesores.

Para utilizar los *shortcodes* solo debe escribirlos en el editor de la página, tal y como se muestra en la siguiente imagen:



Cuando el usuario quiera visitar esta página, se le mostrará el contenido invocado por dicho *shortcode* de la siguiente forma:

Ocupación diaria

Estas son las reservas programadas para el 27/04/2015

Hora	AulaGlobal	Aula 1	Aula 2	Aula 3	Aula 4	Aula 5
09:00						
09:15						
09:30						
09:45		Asignatura 1 (G01)				Asignatura 3 (G11)
10:00		Grado 1				Grado 3
10:15						
10:30						
10:45						
11:00						
11:15						
11:30						
11:45						
12:00						
12:15						
12:30						
12:45						
13:00						
13:15						
13:30						
13:45						
14:00						
14:15						
14:30						
14:45						
15:00						
15:15						
15:30						
15:45						
16:00						
16:15						
16:30						
16:45						
17:00						
17:15						
17:30		Arquitectura de Computadores (G03)				
17:45		Grado en Ingeniería Informática				
18:00						
18:15						
18:30						
18:45						
19:00						
19:15						
19:30						
19:45						
20:00						
20:15						
20:30						
20:45						

RESERVAS PUNTUALES

Asignatura	Profesor	Aula	Horario
Arquitectura de Computadores (G03)	OSCAR PEREZ ALONSO	Aula 1	De 17:00 a 19:00

Ocupación

- Ocupación diaria
- Aula 2.1.G03
- Aula 2.2.G03
- Aula 4.0.F16
- Aula 4.0.F18

Esta semana

- Lunes
- Martes
- Miércoles
- Jueves
- Viernes

Semana próxima

- Lunes de la semana que viene
- Martes de la semana que viene
- Miércoles de la semana que viene
- Jueves de la semana que viene
- Viernes de la semana que viene

Menú Principal

- Ocupación de las aulas
- Horario de las Aulas
- Apertura de cuentas
- Tutoriales
- P.A.Q.

Última Hora

[Sample Post 6 15 mayo, 2015](#)
[Sample Post 5 15 mayo, 2015](#)
[Sample Post 4 15 mayo, 2015](#)

El gestor de reservas tiene los siguientes *shortcodes* disponibles para su utilización, tal y como se detalla en la sección de ayuda incluida en el menú de opciones de la aplicación:

Ayuda

Shortcodes

Para el correcto uso del plugin es necesaria la utilización de códigos abreviados o *shortcodes* de Wordpress. Estos códigos sirven para invocar funciones que ejecuten distintas funcionalidades del plugin y muestren el contenido del mismo a los usuarios. Los distintos *shortcodes* que puede utilizar con este plugin son:

Reservas de hoy	[reservas]
Reservas del día	[reservas dia="(Lunes),(Martes),etc"]
Reservas por aula	[reservas aula="nombre aula"]
Panel de usuario	[login]

Bibliografía

Debian – El sistema operativo universal [en línea] (s.f.). Disponible en: <https://www.debian.org/>

Drupal – Open Source CMS [en línea] (s.f.). Disponible en: <https://www.drupal.org/>
Drupal.org [en línea]. Scheduler [Consulta: 16 de marzo 2015]. Disponible en: <https://www.drupal.org/project/scheduler>

Calderón Mateos, Alejandro. 2014. *Guía Documentación*.

Joomla! The CMS Trusted By Millions for their Websites [en línea] (s.f.). Disponible en: <http://www.joomla.org/>

Joomla! Extension Directory [en línea]. Scheduler [Consulta: 16 de marzo 2015].
Disponible en: <http://extensions.joomla.org/extension/scheduler>

Joomla! Extension Directory [en línea]. Schedule School [Consulta: 16 de marzo 2015].
Disponible en: <http://extensions.joomla.org/profile/extension/living/education-a-culture/schedule-school>

jTable.org [en línea] (s.f.). Disponible en: <http://www.jtable.org/>

Netbeans Plugin [en línea]. PHP WordPress Blog/CMS [Consulta: 16 de marzo 2015].
Disponible en: <http://plugins.netbeans.org/plugin/46542/php-wordpress-blog-cms>

Óscar Rodríguez Calero. "Comparativa entre gestores de contenidos (CMS)" [blog].
[Consulta: 16 de marzo 2015]. Disponible en: <http://www.hiberus.com/blog/comparativa-entre-gestores-de-contenidos-cms>

Tienda de informática online: venta de Ordenadores, móviles y más [en línea] (s.f.).
Disponible en: <http://www.pccomponentes.com/>

Tusalarario.es [en línea]. Calcula y Compara tu Suelo en España. [Consulta: 27 de abril 2015]. Disponible en: <http://www.tusalarario.es/main/salario/comparatusalarario>

Welcome to NetBeans [en línea] (s.f.). Disponible en: <https://netbeans.org/>

WordPress [en línea] (s.f.). Disponible en: <https://wordpress.org/>

WordPress [en línea]. WordPress > Guidelines [Consulta: 5 de mayo 2015]. Disponible en: <https://make.wordpress.org/themes/handbook/guidelines/>

WordPress Codex [en línea]. Writing a Plugin [Consulta: 20 de mayo 2015]. Disponible en: https://codex.wordpress.org/Writing_a_Plugin

WordPress Plugins [en línea]. WordPress > The Events Calendar [Consulta: 16 de marzo 2015]. Disponible en: <https://wordpress.org/plugins/the-events-calendar/>

WordPress Plugins [en línea]. WordPress > Weekly Class Schedule [Consulta: 16 de marzo 2015]. Disponible en: <https://wordpress.org/plugins/weekly-class-schedule/screenshots/>

Acrónimos, abreviaturas y definiciones

AJAX: Técnica de desarrollo web para crear aplicaciones interactivas.

Apache: Servidor web HTTP de código abierto.

API: Interfaz de Programación de Aplicaciones.

CMS: *Content Management System* o Sistema de Gestión de Contenidos.

CSS: Lenguaje de estilo para desarrollo web.

ECTS: Sistema Europeo de Transferencia y Acumulación de Créditos.

Excel: Software incluido en la suite de Microsoft Office.

Feedback: Retroalimentación.

GPL: Licencia Pública General de GNU.

HDD: Disco Duro.

HTML: Lenguaje de programación web.

IDE: Ambiente de Desarrollo Integrado.

ISS8: *Internet Information Services*.

Javascript: Lenguaje de programación web del lado del cliente.

jQuery: Biblioteca de Javascript.

LDAP: Protocolo Ligero/Simplificado de Acceso a Directorios.

LDI: Laboratorio del Departamento de Informática.

Máquinas Virtuales: Software que simula una computadora.

MongoDB: Sistema de base de datos noSQL.

MVC: Modelo Vista Controlador.

MySQL: Sistema de base de datos.

NetBeans: IDE de desarrollo web.

PHP: Lenguaje de programación web del lado del servidor.

RAM: Memoria de Acceso Aleatorio.

Responsive: diseño web que se adapta automáticamente a móviles y tablets.

Shortcode: Conjunto de funciones para la creación de códigos macro para su uso en los post y páginas de Wordpress.

Stakeholder: grupo o personas interesadas en el desarrollo de la aplicación.

SQLite: Versión ligera de MySQL.

SQLServer: Sistema de base de datos.

TFC: Trabajo Final de Grado.

Ubuntu: sistema operativo en base Linux de código abierto.

Windows Server: versión de Windows para servidores.

XEN: Monitor de máquina virtual de código abierto.

- Fin del documento -